NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

APPLIED CYBER OPERATIONS CAPSTONE PROJECT REPORT

IDENTIFICATION AND TRIAGE OF COMPROMISED VIRTUAL MACHINES

by

John Paulenich
Chukwuemeka Agbedo
Kenneth Rea

September 2014

Capstone Advisor:	Garrett McGrath
Co-Advisor:	Scott Cote

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2014	**3. REPORT TYPE AND DATES COVERED** Capstone Report

4. TITLE AND SUBTITLE IDENTIFICATION AND TRIAGE OF COMPROMISED VIRTUAL MACHINES	**5. FUNDING NUMBERS**
6. AUTHOR(S) John Paulenich, Chukwuemeka Agbedo, Kenneth Rea	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The increasing volume and sophistication of cyber-attacks, the adoption of virtualization technology, and the slow incorporation of new software on Navy networks has created a unique situation. The status quo has left those responsible for administering and defending Navy networks at a distinct disadvantage. They are unable to leverage current triage tools available to assist in the identification, classification, and recovery aspects of incident response on a computer network. At the same time, their adversaries have no such limitations. This capstone report explores the use of native operating system tools along with mirrored domains in a virtualized environment as a possible strategy to provide these capabilities.

For this project, we created a generalized virtual network with mirrored domains. In this environment, we developed a toolkit, comprised of software already available to administrators, and a method for deploying it. We then demonstrated its efficacy in detecting a compromise by inserting malware into a computer in the environment. Finally, we used the mirrored domains within the environment to provide a means for an accelerated recovery. Used together, this native toolset and recovery strategy provide a possible solution for the detection of and response to incidents on a network.

14. SUBJECT TERMS Remote Triage, Virtualization, Native Tools, Mirrored VLAN, ESXi, Sysinternals	**15. NUMBER OF PAGES** 117
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

IDENTIFICATION AND TRIAGE OF COMPROMISED VIRTUAL MACHINES

John Paulenich
Chukwuemeka Agbedo
Kenneth Rea

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED CYBER OPERATIONS

from the

**NAVAL POSTGRADUATE SCHOOL
September 2014**

Authors:	John Paulenich Chukwuemeka Agbedo Kenneth Rea
Approved by:	Garrett McGrath Project Advisor
	Scott Cote Co-Advisor
	Cynthia Irvine Chair, Cyber Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The increasing volume and sophistication of cyber-attacks, the adoption of virtualization technology, and the slow incorporation of new software on Navy networks has created a unique situation. The status quo has left those responsible for administering and defending Navy networks at a distinct disadvantage. They are unable to leverage current triage tools available to assist in the identification, classification, and recovery aspects of incident response on a computer network. At the same time, their adversaries have no such limitations. This capstone report explores the use of native operating system tools along with mirrored domains in a virtualized environment as a possible strategy to provide these capabilities.

For this project, we created a generalized virtual network with mirrored domains. In this environment, we developed a toolkit, comprised of software already available to administrators, and a method for deploying it. We then demonstrated its efficacy in detecting a compromise by inserting malware into a computer in the environment. Finally, we used the mirrored domains within the environment to provide a means for an accelerated recovery. Used together, this native toolset and recovery strategy provide a possible solution for the detection of and response to incidents on a network.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AD	Active Directory
ADS	Alternate Data Stream
AFNIC	Air Force Network Integration Center
APT	Advanced Persistent Threat
ASIC	Application Specific Integrated Chip
BIND	Berkeley Internet Name Domain
CANES	Consolidated Afloat Network Enterprise Services
CCIE	Cisco Certified Internetwork Expert
CFI	Canonical Format Indicator
COOP	Continuity of Operations Plan
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
DNS	Domain Name Service
DOD	Department of Defense
DON	Department of the Navy
DRAAS	Disaster Recovery as a Service
DSRM	Directory Services Restore Mode
ENISA	European Union Agency for Network and Information Security
ESW	Ethernet Switch
GB	Gigabyte
GIG	Global Information Grid
GNS3	Graphic Network Simulator v3
GUI	Graphical User Interface
HFS	Hierarchical File System
HKLM	HKEY Local Machine
IAM	Information Assurance Manager
IEEE	Institute of Electrical and Electronic Engineers
IIS	Internet Information Services
ISL	Inter-Switch Link
ISNS	Integrated Shipboard Network System

IOC	Indicator of Compromise
IOS	Internetwork Operating System
IP	Internet Protocol
IT	Information Technology
JNCIE	Juniper Networks Certified Internet Expert
KMS	Key Management System
MB	Megabyte
MD5	Message Digest 5
MIR	Mandiant Intelligent Response
MSI	Microsoft Software Installer
MUI	Multilingual User Interface
NFS	Network File System
NIC	Network Interface Card
NM	Network Module
NMCI	Navy and Marine Corps Intranet
NTC	Navy Tactical Cloud
NTFS	New Technology File System
OS	Operating System
PID	Process Identifier
POR	Program of Record
RAM	Random Access Memory
RAT	Remote Access Tool
RETRI	Rapid Enterprise Triage
RIP	Routing Information Protocol
SHA	Secure Hashing Algorithm
TPID	Tag Protocol Identifier
USB	Universal Serial Bus
USSS	United States Secret Service
VID	VLAN Identifier
VLAN	Virtual Local Area Network
VM	Virtual Machine

I. INTRODUCTION

The Department of the Navy (DON) is notoriously slow at updating and implementing new software in its computer networks. This is evidenced by the fact that the Navy has a plan in place to continue using Microsoft's Windows XP on its networks until the year 2017 [1]. This extends the use of a single Operating System (OS) to over sixteen years. This trend applies to security software as well. The current Department of Defense (DOD) certification and accreditation process is not poised to utilize many current products.

There is one major change on the horizon for Navy networks, and that is virtualization. The introduction of the Navy Tactical Cloud (NTC) and Consolidated Afloat Network and Enterprise Services (CANES) both leverage virtualization technology to help in the streamlining of operations and reduction of costs. Together these represent the future of Navy networking and a fundamental shift in network infrastructure. Space and Naval Warfare Systems Command (SPAWAR) recently issued contracts for the production phase of CANES spanning eight years and amounting to 2.5 billion dollars [2].

While Navy networks are slow to change, attackers are constantly leveraging newer technology and practices. Current trends show that new malware is being discovered at an escalating rate [3]. Along with the growing volume of attacks, the increasing level of sophistication of these attacks is another cause for concern. According to a Frost and Sullivan white paper, many of the newer compromises are generated by state-sponsored and criminal organizations using long term, multistage attacks known as Advanced Persistent Threats (APT) [4].

The previously discussed areas present a problem. The combination of the slow adoption of new software, along with the rapid evolution of malware leaves those in charge of managing Navy networks at a distinct disadvantage. They are required to ensure the security of Navy information Systems (IS), but are denied some of the most powerful tools available to do so. At the same time, the introduction of virtualization provides more opportunities and flexibility in network configuration. It is with these issues in mind that we designed this project.

A. OBJECTIVES

We had several objectives in mind when undertaking this project. For one, we wanted to create a reproducible enterprise network environment. Within this environment we wanted to develop a toolset available to Navy personnel for identifying and classifying compromises in the security of computers on the network. While doing so, we kept in mind the lack of specialized software available to Navy network administrators. We wanted to demonstrate how these tools could be used to efficiently detect the presence of a compromise on the network. Once the compromise has been determined, we wanted to present a possible recovery strategy using mirrored domains and Virtual Local Area Networks (VLAN) when responding to an incident on the network.

B. METHODOLOGY

In order to meet our objectives we took the following steps. We performed a review of some of the available products used in computer triage. This was done to examine the capabilities that were currently available and to determine what information our toolset could provide. From there we began to design and implement our scaled-down enterprise network. We utilized an ESXi server cluster to create a completely virtualized network. Inside of this network environment we assembled a toolkit comprised of operating system executables and software from the Microsoft Sysinternals suite. After that, we then inserted malware onto a host in the network to simulate a compromise. We then employed these tools to gather information relevant for detecting a compromise from all of the machines in the network remotely. Next, we used the information gathered from the tools to provide the evidence and nature of the compromise. While doing so we also provided an analytical framework for examining the output of the tools. Finally, with the compromise of the network determined, we demonstrated the capability of mirrored VLANs as a viable recovery strategy.

C. BENEFITS

This project provides two significant benefits. First it provides a toolset that can be employed by administrators on Navy networks. Since it only utilizes software packaged with the operating system and the Sysinternals suite, it is readily available on all systems. We also created a means for deploying them remotely, expediting the process of identifying and classifying incidents on a network. This project also provides a tested strategy that could be used to aid in the expedited recovery of compromised computers within a virtual network.

D. REPORT STRUCTURE

Chapter I presents the objectives and scope of this project. It provides an overview of the methods that were used in the development and implementation of the test environment. Finally, it states the benefit of this project to both the DON and the DOD as a whole.

Chapter II is a review of triage implementations currently in use for security compromises. It examines the processes and tools being used in the corporate world. Examples include Mandiant's Intelligent Response (MIR), and Palo Alto Network's WildFire. Next it reviews how entities in the government, specifically the DON, are performing triage on their networks. Finally, it describes a very specific form of triage known as Rapid Enterprise Triaging (RETRI), which was the springboard for the development of this project.

Chapter III defines the virtualized environment in which this project was accomplished. It provides a network schematic defining the domains and forest structure. It describes the numbers of machines used, their OS, configuration, and services provided. It also defines the setup and configuration of the routing and switching in the ESXi environment.

Chapter IV details the tools used in the detection and classification portion of this project. It looks closely at many of the native operating system tools and their expected outputs. It also reviews additional tools such, Microsoft Sysinternals, and their ability to shed further light on potential security compromises of the systems on a virtualized network through the comparison of baseline outputs with compromised outputs. Finally, the chapter explains the methods we used to employ these tools.

Chapter V examines the malware that was used to create the compromise on the network. We look at the background of the malware and define its capabilities. It also covers the server and client portions of the malware. Finally, it provides the details of our specific implementation of the malware.

Chapter VI provides an in-depth analysis of the results obtained through the use of the defined tools. It compares and contrasts the baseline outputs with those outputs obtained after the malware is inserted. This defines the indicators that personnel should be able to recognize in determining if there has been a network compromise. Finally, it demonstrates the capability of VLAN mirroring to allow a rapid recovery from a compromise, while still enabling examination of the compromised assets.

Chapter VII summarizes our project. It outlines what we have accomplished and the areas of work that remain for future research.

II. CURRENT TRIAGE PRODUCTS

In order to ensure our toolkit provided the necessary capabilities to identify and classify compromises in a network we needed to examine the idea of computer triage. In addition to the conceptual review, we researched some of the current products available providing triage capabilities to network administrators and security professionals in order to guide the development of our toolset.

A. TRIAGE

The term *triage* is most often used in the medical field. It is the assignment of degrees of urgency to wounds or illnesses to decide the order of treatment of a large number of casualties. Computer security has recently adopted the term as well. The European Union Agency for Network and Information Security (ENISA) includes triage as one of the phases in the incident handling process, and further divides triage into three sub-phases: verification, initial classification and assignment [5]. Triage is most often used to determine which systems require the most urgent action, when there are multiple computers on a network that have been compromised. This provides a framework for incident responders to determine a course of action when responding to the compromise.

B. LOCAL TRIAGE

The American cyber security firm Mandiant provides remote forensics and incident response services. For local capabilities, i.e., those requiring physical access to a potentially compromised machine, Mandiant offers Redline, a free tool enabling an analyst to discover Indicator of Compromise (IOCs) via memory and file analysis [6]. Figure 1 provides an illustration of the Redline process. Mandiant recommends that its Redline be used via a physically connected USB flash drive that is moved from machine to machine. This tool used on its own is an example of local triage in that it allows for an assessment of a computer’s security status, but it requires physically accessing each machine to do so. It works well for a local network with a small numbers of computers to maintain and investigate. However, physically traveling from system to system to conduct such an assessment would be time consuming and costly, and is impractical in most enterprise network environments. Also, given the inherent risk of migrating malware from host to host via USB droppers, this method of surveying devices could not only violate network policies, but could also spread malware.

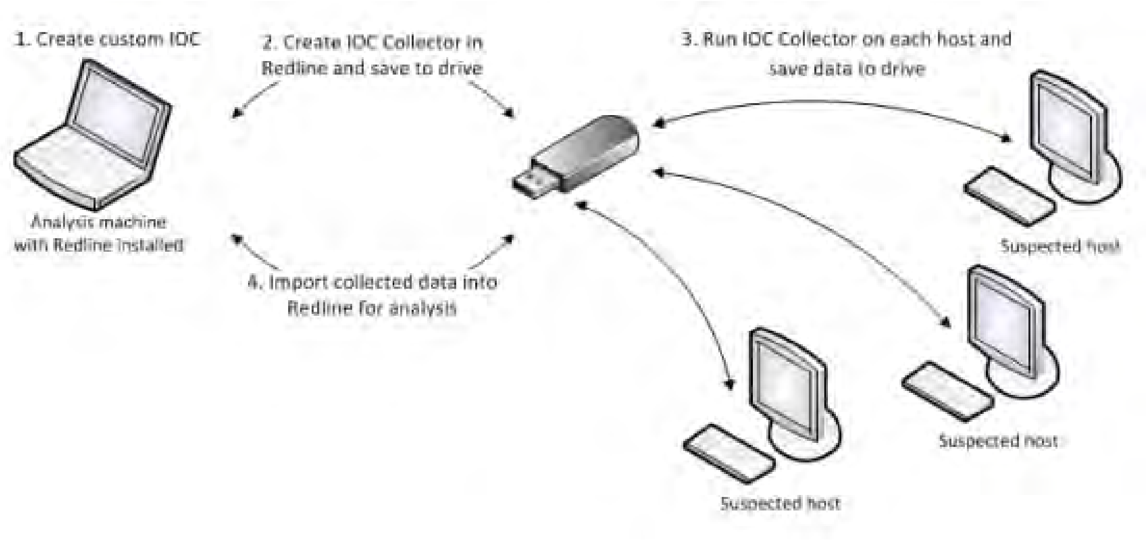

1. Create custom IOC
2. Create IOC Collector in Redline and save to drive
3. Run IOC Collector on each host and save data to drive
Analysis machine with Redline installed
4. Import collected data into Redline for analysis
Suspected host
Suspected host
Suspected host

Figure 1. Mandiant Redline tool overview, from [6].

C. REMOTE TRIAGE

The next step in the evolution of computer triage is remote triage. Remote Triage is a security analysis technique used by incident responders to investigate potential unauthorized access and other anomalous behavior on endpoint systems within a network. The key aspect of remote triage is effectively assessing the security status of a computer without having to physically visit the machine. Remote triage often involves the use of software which is centrally located on a network resource and is able to be deployed onto client machines to automatically gather information. This data is then retrieved and processed by a dedicated analysis machine [7]. The remainder of this chapter presents some of the current implementations of remote triage in both the commercial sector and the Department of Defense.

1. Mandiant

In addition to the free Redline tool, Mandiant offers the Mandiant Intelligent Response (MIR) service. This gives customers the ability to remotely investigate client devices on their network [8]. While these two tools can be used in conjunction, to use Redline remotely requires MIR services. The Mandiant website does not elaborate on the underlying technical mechanics of the capabilities of these products. Mandiant Intelligent Response and Redline combined is one example of remote triage being used in the private sector.

2. Palo Alto Networks

Another example of remote triage is Palo Alto Network's WildFire. Several features distinguish this product. Detection and triage are completely automated within the WildFire software. WildFire resides on dedicated hardware and has access to their cloud-based service which can be deployed privately within the customer's network or accessed over the Internet. Rather than using signatures and predefined actions to diagnose malware, WildFire uses behavioral analysis. Suspicious software is sent to a cloud-based sandbox capable of detecting over one hundred malicious behaviors. Once the malware is classified, WildFire generates protections to block the threat and shares this data with all WildFire subscribers across the globe within one hour [9]. This results in an expedited process of detection, diagnosis and correction. Figure 2 illustrates the WildFire process in handling the discovery of malware.

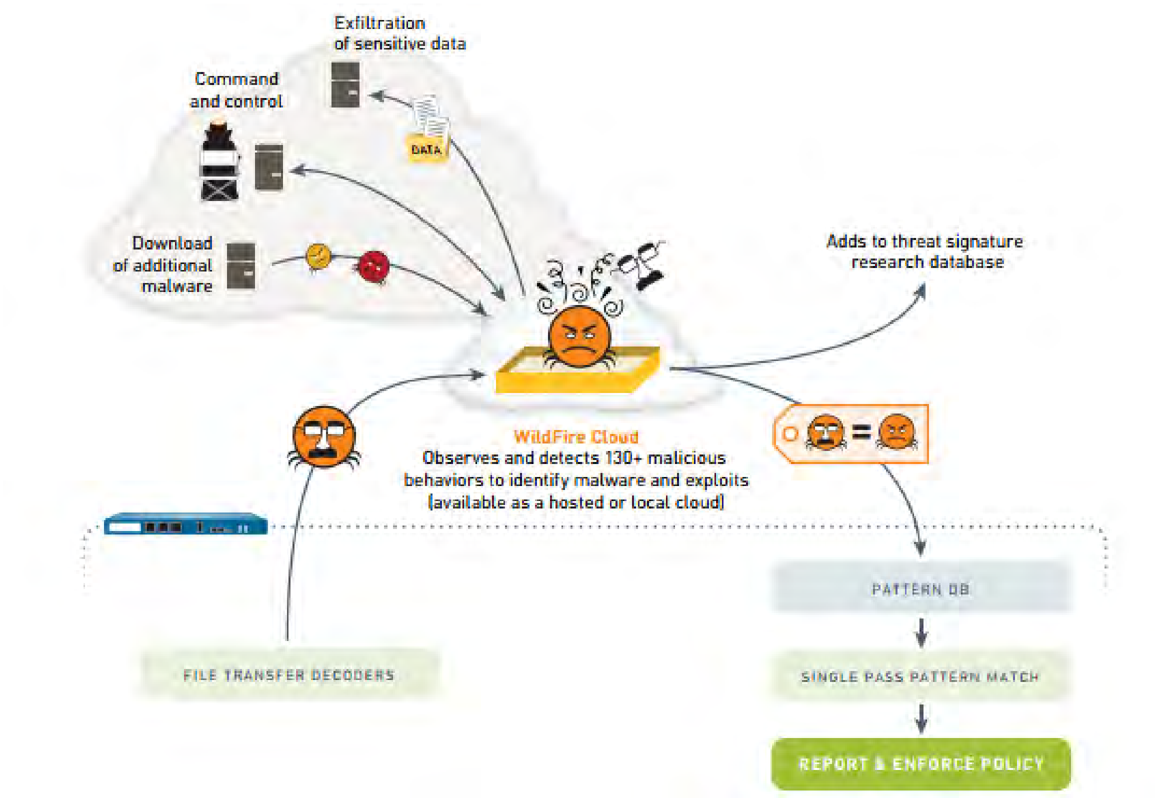

Exfiltration of sensitive data
Command and control
DATA
Download of additional malware
Adds to threat signature research database
WildFire Cloud
Observes and detects 130+ malicious behaviors to identify malware and exploits (available as a hosted or local cloud)
PATTERN DB
FILE TRANSFER DECODERS
SINGLE PASS PATTERN MATCH
REPORT & ENFORCE POLICY

Figure 2. WildFire process from detection to correction, from [10]

3. Department of Defense

The United States Department of Defense developed a remote triage tool suite known as Blue Scope. It is deployed on a laptop physically connected to the network being triaged, which allows the security analyst to remotely access every endpoint device on the LAN. Remote triage with Blue Scope is conducted by remotely connecting and performing a series commands on an endpoint system. These commands construct a picture of the internal operations of each system being interrogated.

4. RETRI

Former Mandiant employees Aaron LeMasters and Michael Murphy gave a presentation titled *RETRI: Rapid Enterprise Triaging* at Black Hat. It detailed the capabilities and implementation of their Codeword tool [11]. Although similar to the Mandiant Intelligent Response, Codeword uses Microsoft Software Installers (MSI) to deploy what they term agents for remote triage purposes. These agents are software packages that monitor the computer's status and permit remote access on each of the computers. They also enable the remote execution of forensic tools across a network's devices [12].

The main distinguishing feature of the RETRI concept though, is its network design. It requires the presence of two fully functional networks: one for production and one for quarantine. Once a system is compromised it is moved from the production to the quarantine network. This process requires a large initial investment of time and resources. However, it reduces the time to recover and restore operations after a compromise. In a virtualized network, such as the one built for this capstone, this could be implemented with fewer resources, further reducing the recovery time. Figure 3 shows the proposed network setup from their Black Hat brief for the RETRI concept.

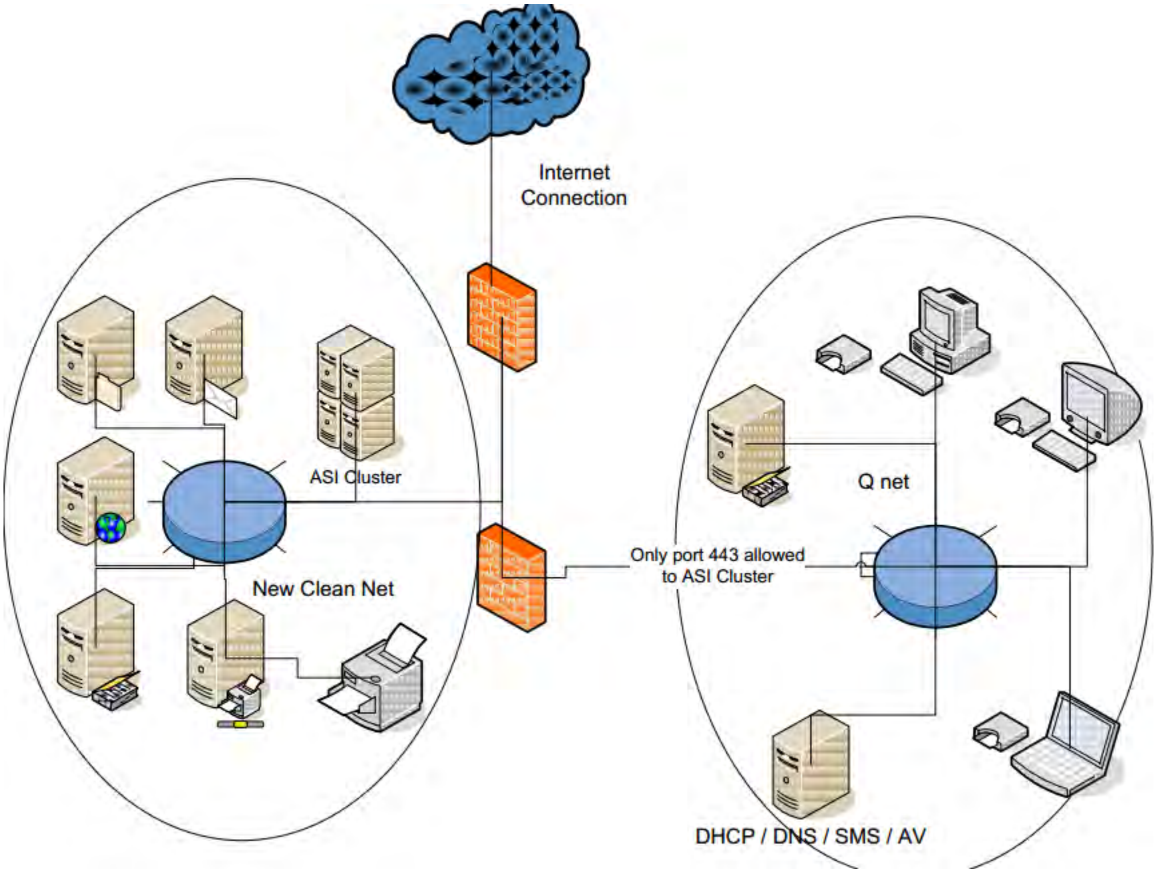


Figure 3. RETRI network topology, from [11]

THIS PAGE INTENTIONALLY LEFT BLANK

III. LAB ENVIRONMENT

The RETRI network design provides an interesting strategy for the accelerated recovery of assets from an incident. We employed this network design in our scenario environment to demonstrate this recovery capability after the initial identification of a compromise using our toolset. This chapter describes the sandbox environment we created for our testing purposes. The environment for this lab exercise is completely virtualized, which provides a simplified means to utilize the mirrored VLAN response and recovery strategy proposed in the RETRI brief. All of the client machines, servers, routers, and switches are run within the ESXi server environment. The network was designed to emulate Navy networks and supports a variety of operating systems and versions. There are four virtual machines used to provide the routing and switching infrastructure for the entire network. There are also four networks, 3 Active Directory domains and 1 Linux network utilizing Samba.

Each production domain also has a mirrored domain to support the expedited recovery. The mirrored domain is identical to the production domain. The only difference between the production domain and the mirror domain is that they reside on different VLANs. After creation, the computers in the mirrored domains are powered off until they are needed in the recovery process. With the discovery of a compromise, machines can be powered on and moved from the mirrored domain to the production domain, quickly restoring basic functionality. Upon completion of installation and setup, the general topology of the network is illustrated in Figure 4.

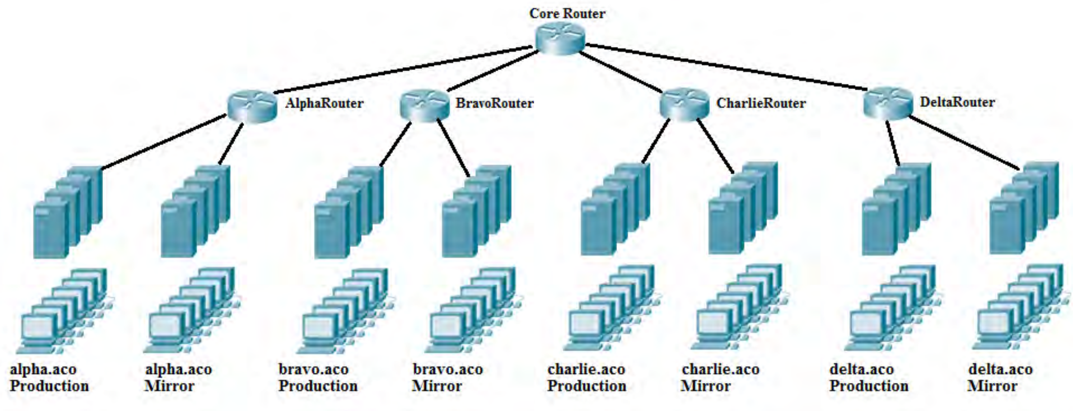


Figure 4. General network topology

A. DOMAIN ALPHA.ACO

The first production domain is alpha.aco. The alpha.aco domain was designed to simulate some of the most up-to-date networks in the DOD and DON. For it, we used a combination of Windows Server 2012 Standard Edition and Windows 7 operating systems. There are a total of ten machines in the domain: four servers and six workstations. Table 1 provides a listing of all of the machines, their role in the network and their IP address.

Table 1. Alpha.aco machine names, roles, and IP addresses

Virtual Machine Name	Role	IP Address
AlphaDC01	Domain Controller DHCP Server DNS Server	10.1.0.21
AlphaFS01	Domain Controller DNS Server File Server	10.1.0.22
AlphaEX01	Mail Server	10.1.0.23
AlphaWWW1	Web Server	10.1.0.24
AlphaWS01	Workstation	10.1.0.25
AlphaWS02	Workstation	10.1.0.26
AlphaWS03	Workstation	10.1.0.27
AlphaWS04	Workstation	10.1.0.28
AlphaWS05	Workstation	10.1.0.29
AlphaWS06	Workstation	10.1.0.30

(1) Servers

In vSphere, each of the four servers in the alpha.aco domain is provisioned with 40GB HD space, 4GB of Random Access Memory (RAM) and one Network Interface Card (NIC). This exceeds the Microsoft recommended values of 32GB HD and 512 MB RAM, and was intended to improve performance in the virtualized environment. All of the server NIC’s for this domain reside in the Alpha VLAN in the vSphere software. ALPHADC01 is the first domain controller, primary Domain Name Service (DNS) server, and Dynamic Host Configuration Protocol (DHCP) server for the domain. Within DHCP, all of the machines on the network have Internet Protocol (IP) address reservations. This ensures that computers on the production domain have the same addressing as computers on the mirror domain. ALPHAFS01 is the primary file share server for the domain with the Network File System (NFS) feature installed. ALPHAEX01 is the mail server for the domain. It uses Microsoft Exchange 2013 as the mail server software. The final server in the domain is ALPHAWWW1, a web server running Microsoft IIS 8. For security these machines are completely patched and updated.

(2) Workstations

For the Windows 7 workstations each virtual machine was provisioned with 20GB HD space, 2GB of RAM and one NIC. All of the NICs for these workstations reside on the ALPHA VLAN as defined in the vSphere software. Though the machines will retain the same IP address, they are still configured for DHCP, to support ease of installation on the client side. Their individual reservations on the DHCP server ensure that their addresses will not change. These machines were left unpatched with automatic updates turned off. We did this in order to simulate new vulnerabilities on systems without the use of zero-day attacks, which is beyond the scope of our project.

B. DOMAIN BRAVO.ACO

The second domain in our enterprise network is the bravo.aco domain. The bravo.aco domain has general similarities to many of the Navy shore station networks. In it we deployed the Microsoft Windows Server 2008 Standard Edition and Microsoft

Windows 7 operating systems. There are a total of ten computers in the bravo.aco domain; 4 servers and six workstations. Table 2 provides a list of the machine names, roles, and IP addresses.

Table 2. Bravo.aco machine names, roles and IP addresses

Virtual Machine Name	Role	IP Address
BravoDC01	Domain Controller DHCP Server DNS Server	10.2.0.21
BravoFS01	Domain Controller DNS Server File Server	10.2.0.22
BravoEX01	Mail Server	10.2.0.23
BravoWWW1	Web Server	10.2.0.24
BravoWS01	Workstation	10.2.0.25
BravoWS02	Workstation	10.2.0.26
BravoWS03	Workstation	10.2.0.27
BravoWS04	Workstation	10.2.0.28
BravoWS05	Workstation	10.2.0.29
BravoWS06	Workstation	10.2.0.30

(1) Servers

In vSphere, each of the four servers in the bravo.aco domain is provisioned with 30GB HD space, 4GB of RAM and one NIC. This exceeds the Microsoft minimum values of 10GB HD and 512 MB RAM, and is intended to help improve the performance in the virtualized environment. All of the server NIC's for this domain reside in the Bravo VLAN. BRAVODC01 is the first domain controller, primary DNS server, and DHCP server for the domain. Within DHCP, all of the machines on the network have specific IP addresses reserved. This is to ensure that computers on the production domain have the same addressing as their counterparts on the mirror domain. BRAVOFS01 is the primary file share server for the domain with the NFS feature installed. It is also a domain controller and secondary DNS server for the domain. BRAVOEX01 is the mail server for the domain. It uses Microsoft Exchange Server 2010 as the mail server software. The final server in the domain is BRAVOWWW1, a web server running Microsoft IIS 7. For security reasons these machine are completely patched and updated.

(2) Workstations

The workstations in the bravo.aco domain are using the Windows 7 Operating System (OS) and are configured in the same manner as those listed in the alpha.aco domain.

C. DOMAIN CHARLIE.ACO

The third domain in our enterprise network is the charlie.aco domain. The charlie.aco domain emulates the majority of Integrated Shipboard Network Services (ISNS) environments using the Microsoft Windows Server 2003 and Microsoft Windows XP operating systems. These are older systems in which much of the software has reached the end of its life cycle, thus creating security challenges for system administrators and Information Assurance Managers (IAM). Table 3 provides a list of computers in the charlie.aco domain with machine names, roles, and IP addresses.

Table 3. Charlie.aco machine names, roles, and IP addresses

Virtual Machine Name	Role	IP Address
CharlieDC01	Domain Controller DHCP Server DNS Server	10.3.0.21
CharlieFS01	Domain Controller DNS Server File Server	10.3.0.22
CharlieEX01	Mail Server	10.3.0.23
CharlieWWW1	Web Server	10.3.0.24
CharlieWS01	Workstation	10.3.0.25
CharlieWS02	Workstation	10.3.0.26
CharlieWS03	Workstation	10.3.0.27
CharlieWS04	Workstation	10.3.0.28
CharlieWS05	Workstation	10.3.0.29
CharlieWS06	Workstation	10.3.0.30

(1) Servers

In vSphere, each of the four Windows 2003 servers in the charlie.aco domain is provisioned with 20GB HD space, 2GB of RAM and one NIC. This exceeds the Microsoft minimum values of 2GB HD and 128 MB RAM, and is intended to help improve the performance in the virtualized environment. All of the server NICs for this domain reside in the CHARLIE VLAN in the vSphere software. We used the same procedures for DHCP that we used in the other domains to ensure clients maintain the same IP address. CHARLIEEX01 is the mail server for the domain. It uses Microsoft Exchange Server 2007 as the mail server software. The final server in the domain is CHARLIEWWW1. It has Microsoft IIS 6 installed to provide the web server capability. For security and functionality reasons these machine are completely patched and updated.

(2) Workstations

For the six Windows XP workstations in the charlie.aco domain each virtual machine was provisioned with 20GB HD space, 2GB of RAM and one NIC. All of the NIC's for these workstations reside on the CHARLIE VLAN in the vSphere software. Though the machines will retain the same IP address, they are still configured for DHCP to support ease of installation on the client side. Their individual reservation on the DHCP server ensures that their address will not change. These machines were left unpatched with automatic updates turned off.

D. DOMAIN DELTA.ACO

The fourth and final domain in our enterprise network is the delta.aco domain. This domain represents non-standard system found throughout the DOD Global Information Grid (GIG). These machines use the Ubuntu distribution of the Linux operating system, as it has many similarities to the HP-UX and Solaris OSs used in certain DON systems, but allows us to avoid licensing costs. Table 4 provides a list of computers in the delta.aco domain with machine names, roles, and IP addresses.

Table 4. Delta.aco machine names, roles, and IP addresses

Virtual Machine Name	Role	IP Address
DeltaDC01	Domain Controller DHCP Server DNS Server	10.4.0.21
DeltaFS01	Domain Controller DNS Server File Server	10.4.0.22
DeltaEX01	Mail Server	10.4.0.23
DeltaWWW1	Web Server	10.4.0.24
DeltaWS01	Workstation	10.4.0.25
DeltaWS02	Workstation	10.4.0.26
DeltaWS03	Workstation	10.4.0.27
DeltaWS04	Workstation	10.4.0.28
DeltaWS05	Workstation	10.4.0.29
DeltaWS06	Workstation	10.4.0.30

(1) Servers

There are a total of four servers in the delta.aco domain. Each uses Ubuntu 12.04 with GUI mode for the operating system. To provide directory services that can be used with the Windows system, the server used as a domain controller in delta.aco, DELTADC01, has Samba installed. The email services are provided to the domain with Sendmail 8.14.9 mail server software. The web server uses Apache 2.4 to provide a public facing website. DNS was implemented on the servers DELTADC01 and DELTAFS01 by installing and configuring the Berkeley Internet Name Domain (BIND) software package. A DHCP server was installed on DELTADC01. We did this by downloading and installing the DHCP software package and then making the necessary changes to the dhcp.conf file.

(2) Workstations

There are six workstations in the delta.aco domain. All of the workstations use Ubuntu 12.04 as their operating system. Each of the workstations was provisioned with

20GB of hard disk space and 2GB of RAM. The workstations in the production domain are part of the vSphere defined DELTA VLAN in the ESX environment, while the workstations in the mirror domain are part of the vSphere defined DELTA MIRROR VLAN. All of the workstations are configured for DHCP but each has a specific reservation on the DHCP server to ensure they will always have the same IP address and to allow the full functionality of the system to continue uninterrupted if it is moved to the mirror domain.

E. INFRASTRUCTURE

An integral part of this lab environment is the routing and switching infrastructure. Rather than using the built-in capabilities for routing and VLAN switching available in vSphere we used another approach which required more software and configuration but allowed for more granular control and is more similar to a real world enterprise environment. To do this, we initially installed and configured four Virtual Machines (VM) with Windows XP. Then we installed the Graphic Network Simulator version 3 (GNS3) software package on each machine to provide routing and switching support.

GNS3 is a suite of software that includes: Dynamips, VirtualBox, and Qemu. It is an open source software project developed to allow for the virtualization of large-scale routing environments. Its initial purpose was to prepare students for portions of the Cisco Certified Internetworking Expert (CCIE) and Juniper Networks Certified Internet Expert (JNCIE) exams without incurring the cost of acquiring the necessary hardware. Now it is also used as a sandbox environment for testing changes to routing features prior to their implementation in production environments. For our lab scenario we primarily used the Dynamips software to support the virtualization of a Cisco Internetworking Operating System (IOS) device on a Personal Computer (PC). Currently GNS3 does not support the virtualization of Cisco Catalyst switches due to the nature of the ASIC processors found in those devices [13].

However, we were unable to achieve full functionality using the GNS3 software. To connect the router simulated on a computer to outside devices requires the use of a bridged loopback adapter. This works well when there is only one device connected to the router. However, when trying to connect the router to multiple machines in a vSphere VLAN the connectivity was intermittent. We contacted the software developers who informed us that it was not intended to be used in such a way and could not offer support for improving the functionality. Our alternative was to use a DD-WRT implementation for our routing and switching. DD-WRT is open source firmware for Linksys routers that allows for more control on typical plug and play routers. The following sections describe our implementation of the DD-WRT infrastructure.

1. VM

For simplicity of operation and configuration, the VM's to be used for the basis of the routing and switching devices were installed as x86 DD-WRT. This variety of DD-WRT software is a Linux-based operating system that can function on standard PC's instead of just Linksys routers. We used the 24461 release, retrievable online from the DD-WRT website [14]. The machines are configured with 10GB of HD, 1GB of RAM, and 3 NICs. The procedures we used configuring and installing the DD-WRT operating system are found in Appendix A.

2. Routing

To further demonstrate the enterprise-level design of the lab environment, we used local routers for each domain and a core router providing interconnectivity. To support the edge routing for each production domain and its corresponding mirrored domain, we installed and configured four routers with three NIC's each, one Wide Area Network (WAN) interface and two Local Area Network (LAN) interfaces. A fifth router was installed and configured for the core routing in between the routers for each domain with two NIC's, one WAN and one LAN. The routers are configured to use Routing Information Protocol (RIP) version 2. Table 5 shows each of the routers, the host VM on which they reside, IP address, subnet mask, and to what they are connected. The procedures for configuring the routers are found in Appendix A.

Table 5. Router connectivity

Host VM	Router Name	IP Addresses	Subnet Mask	Connection Point
rtralpha	Alpharouter	10.1.0.1 192.168.1.2	255.255.0.0 255.255.255.0	Alpha.aco Core Router
rtrbravo	Bravorouter	10.2.0.1 192.168.2.2	255.255.0.0 255.255.255.0	Bravo.aco Core Router
rtrcharlie	Charlierouter	10.3.0.1 192.168.3.2	255.255.0.0 255.255.255.0	Charlie.aco Core Router
rtrdelta	Deltarouter	10.4.0.1 192.168.4.2	255.255.0.0 255.255.255.0	Delta.aco Core Router
rtrcore	corerouter	192.168.1.1 192.168.2.1 192.168.3.1 192.168.4.1	255.255.255.0 255.255.255.0 255.255.255.0 255.255.255.0	Alpha Router Bravo Router Charlie Router Delta Router

3. Switching

Switching in the lab environment is accomplished by adding the configuration of VLANs within the DD-WRT interface. This is necessary in order for the environment to be able to utilize 802.1Q. 802.1Q is the IEEE standard for tagging traffic for a particular VLAN on a trunk through the addition of 32 bit tag to the Ethernet header between the source address and type/length fields. Inside the tag are four fields: TPID, Priority, CFI, and VID [15]. For our purposes the most important of these fields is the VID as it correctly identifies the VLAN for which the traffic is destined. This enables us to have each domain completely mirrored, including machine names and IP addresses while maintaining the ability to segregate the traffic between the production domain and the mirror domain.

Also to support the switching environment, each production and mirror domain was given its own VLAN in the vSphere software configuration. Table 6 illustrates the breakout of VLANs by the routers, their label in vSphere, their programmed VLAN number, and which portion of the network they service. The procedures for installing and configuring the network switching are found in Appendix A.

Table 6. Switching breakout

Router Name	vSphere VLAN Label	Switch VLAN Label	Domain Serviced
Alpharouter	ALPHA ALPHA MIRROR	VLAN 11 VLAN 12	Alpha prod. Alpha mirror
Bravorouter	BRAVO BRAVO MIRROR	VLAN 13 VLAN 14	Bravo prod. Bravo mirror
Charlierouter	CHARLIE CHARLIE MIRROR	VLAN 15 VLAN 16	Charlie prod. Charlie mirror
Deltarouter	DELTA DELTA MIRROR	VLAN 17 VLAN 18	Delta prod. Delta mirror

THIS PAGE INTENTIONALLY LEFT BLANK

IV. FORENSIC TOOLSET

The tools we examined in Chapter II represent some of the best capabilities currently available for computer triage. Software of this nature is not installed on Navy networks due to program of record policies. However, many of the tools used in specialized computer network security suites have originated from a native tool delivered with the operating system. Navy networks do have these Windows native OS tools and the Sysinternals suite. In developing our toolkit we wanted to focus on using software allowed on government networks and available to DOD administrators and analysts. These tools can be leveraged to provide a remote triage capability for administrators and analysts on afloat networks. In this chapter we examine the individual pieces of our forensic toolkit. To do so, the individual tools must be assembled into a centralized location, and precautions must be taken to ensure the integrity of the tools themselves. Finally, a method for the deployment of the tools in an automated and efficient manner is presented.

A. EXAMINATION OF INDIVIDUAL TOOLS

1. Native Tools

The native operating system tools we chose each provide a small piece of information; when used together they can provide information on the status of a system as a whole. Table 7 provides a list of the native tools chosen and the information that they can provide. Specific information that could be an IOC for each tool is provided in the analysis section of Chapter 6.

Table 7. Native operating system tools and their capabilities

date.exe	Displays or sets the system date. date /t **:** Displays the current date without prompting for a new date [16]
time.exe	Displays or sets the system time. time /t **:** Displays the current system time, without prompting for a new time [17]
ipconfig.exe	Displays the current Transmission Control Protocol/Internet Protocol (TCP/IP) configuration [18]

net.exe	
net session	Manages server computer connections. Used without parameters, net session displays information about all sessions with the local computer [19]
net use	Used without parameters, net use retrieves a list of network connections [20]
net share	Used without parameters, net share displays information about all of the resources that are shared on the local computer [21]
net start	Used without parameters, net start displays a list of services that are currently operating [22]
sc.exe query	Obtains and displays information about the specified service, driver, type of service, or type of driver [23]
driverquery.exe	Displays a list of all installed device drivers and their properties [24]
tasklist.exe	Displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer [25]. Appendix B provides common processes running in the lab environment and their purposes.
netstat.exe	Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connections [26]
nbtstat.exe	Nbtstat is designed to help troubleshoot NetBIOS name resolution problems. When a network is functioning normally, NetBIOS over TCP/IP (NetBT) resolves NetBIOS names to IP addresses. The nbtstat command removes and corrects preloaded entries using a number of case-sensitive switches. The nbtstat -a < *name* > command performs a NetBIOS adapter status command on the computer name specified by < *name*>. The adapter status command returns the local NetBIOS name table for that computer as well as the MAC address of the adapter card. The nbtstat -A <*IP address* > command performs the same function using a target IP address rather than a name [27]
reg query	Returns a list of the next tier of subkeys and entries that are located under a specified subkey in the registry.
<KeyName>	Specifies the full path of the subkey. For specifying remote computers, include the computer name (in the format \\ComputerName\) as part of the *KeyName*. Omitting \\ComputerName\ causes the operation to default to the local

	computer. The *KeyName* must include a valid root key. Valid root keys for the local computer are: HKLM, HKCU, HKCR, HKU, and HKCC. If a remote computer is specified, valid root keys are: HKLM and HKU [28]
doskey.exe	recalls a history of commands entered via the current command prompt session, edits command lines, and creates macros [29]

2. Sysinternals Suite Tools

The native operating system tools provide general information on the status of a system. However, for a more in-depth view of possible security issues, we chose to use some of the tools from the Microsoft Sysinternals suite. These tools provide additional information about the state of a system not available from the native tools. Table 8 contains a list of the Sysinternals tools we chose and the information they provide.

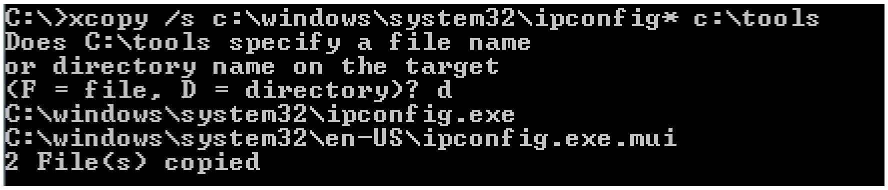
Table 8. Sysinternals tools and their capabilities

psexec.exe	Psexec.exe enables to the execution of programs on remote systems. It encrypts all communication between local and remote systems. [30]
psservice.exe	*PsService* is a service viewer and controller for Windows. Like the sc utility included in the Windows NT and Windows 2000 Resource Kits, *PsService* displays the status, configuration, and dependencies of a service, and allows starting, stopping, pausing, resuming and restarting them. Unlike the SC utility, *PsService* enables users to logon to a remote system using a different account from the one currently they are currently using. This is useful for cases when the account does not have required permissions on the remote system. *PsService* includes a unique service-search capability, which identifies active instances of a service on a network. [31]
psloggedon.exe	Displays locally and remotely logged on users. When specifying a user name instead of a computer, *PsLoggedOn* searches the computers in the domain and displays if the user is currently logged on [32]
logonsessions.exe	Lists the currently active logon sessions and the -p option displays the processes running in each session [33]
pslist.exe	Shows statistics for all the processes. (–t shows process tree) [34]

handle.exe	*Handle* is targeted at searching for open file references, so without command-line parameters it will list the values of all the handles in the system that refer to open files and the names of the files. [35]
autorunsc.exe	Shows the currently configured auto-start applications as well as the full list of Registry and file system locations available for auto-start configuration. [36]
listdlls.exe	ListDLLs is a utility that reports the DLLs loaded into processes. ListDLLs can also display full version information for DLLs, including their digital signature, and can scan processes for unsigned DLLs [37]
sigcheck.exe	Sigcheck is a utility that aids in verification of file integrity. It can be used to hash files with several algorithms, verify file certificate chains, and search for non-signed files in sensitive directories [38].

B. BUILDING THE TOOLKIT

Once we decided upon the list of tools and their intended purposes we created a new virtual machine to serve as our triage workstation for the enterprise network. For this we installed the Microsoft Windows 7 Professional operating system. This was a fully patched and updated system to give higher assurance to the data and software maintained on this machine. We began by gathering the specified tools in a folder named *tools* on the C: drive. The Sysinternals tools were downloaded from the Microsoft website, and the individual tools were placed in this folder. For the native tools, we initially copied them from the c:\windows\system32 directory. However, when attempting to run these tools from c:\tools, they did not work. The software would run, but it provided no output. After some research we discovered that Microsoft uses Multilingual User Interface (MUI) files to store resources for many of the commands in the System32 directory, such as ipconfig, netstat, nbtstat, and tasklist [39]. In order for the executable files to function properly, we needed these resource files to be copied to our toolset location. To do this we used the *xcopy* command with the –s argument to copy subdirectory files as well as the executable. Figure 5 shows the exact command line syntax and the results. This was then repeated for all of the necessary tools in the System32 directory.



```
C:\>xcopy /s c:\windows\system32\ipconfig* c:\tools
Does C:\tools specify a file name
or directory name on the target
(F = file, D = directory)? d
C:\windows\system32\ipconfig.exe
C:\windows\system32\en-US\ipconfig.exe.mui
2 File(s) copied
```

Figure 5. Command to copy .exe and .mui files

Once we found all of the software and placed it in our tools folder we began to test each tool in the different operating system environments to ensure we were getting the expected outputs as described by the tables above. The tools functioned as expected on all of the Windows 7 clients in the enterprise and on the Windows Server 2012 machines. However, when we ran the tools on the Windows XP clients, the Server 2008 machines and the Server 2003 machines we noticed some issues in their output. Many of the System32 tools from a Windows 7 machine would not function in these environments generating errors stating that the programs were not valid Win32 executable files even though they were retrieved from the 32 bit version of the OS. To address this issue we copied the needed executable files from each version of Windows and then created tool folders for each of the operating systems.

1. Tool Integrity

With our toolset assembled, the next step was to ensure their fidelity. Since the tools were collected from newly installed machines in a closed environment we were reasonably sure that the files had not been compromised. However, there is no guarantee that the files would remain untouched. Microsoft does not provide the hash values for their tools. Instead, they state that if the software is downloaded from their site, it is authentic [40]. So, we took hashes of each of the files. To do so we used the sigcheck utility in the Sysinternals suite, which provides a Secure Hashing Algortihm-1 (SHA-1), SHA-256, and Message Digest-5 (MD5) hash values [38]. We then stored these hash values in a separate location on an ISO image to prevent tampering. Prior to using the tools, sigcheck should be run on the tools directory, and the hash values should be compared to known good values. If there are any differences, it is possible the tools have been compromised and their outputs can no longer be trusted. If a compromise to the triage machine is suspected, the sigcheck utility may report no issues when in fact there are. In this case, the hard disk for the triage machine would need to be mounted and examined on a new machine, and the hashes calculated from software on the new machine. Table 9 provides the hash values for the different algorithms output from the sigcheck utility.

Table 9. Summary of hash information provided by sigcheck

Tool	Algorithm	Hash Value
autorunsc.exe	MD5	E6C7AA779C7EBBB53EFE8C8691FF161E
	SHA1	48E938A7B8849A4216C790CCE7D4FC6BDD3BA35D
	SHA256	B6735886CB77284769663BDF06A7F5E4BA564DD8630FC36DCCE57366CD125BDF
driverquery.exe	MD5	E2BCD723EA3517E71A154502127B5D92
	SHA1	4EF626BFC18E4707A195A79A975392B30D0D603E
	SHA256	0E831713C435D85C6FAB664E344742D72177C93F7A21E3187D959C5C58B071CC
handle.exe	MD5	C8AE5979CE001F5FF34AC1D105839C1C
	SHA1	CE715D9677DBB9A56CF07D00B4847A12B5F0ED21
	SHA256	1C99E37E6186EF359902183F746C400C01F04AB8F5442CB2D60F801A617A25B0
ipconfig.exe	MD5	CF45949CDBB39C953331CDCB9CEC20F8
	SHA1	6756F752141602424AF234433DADEDC12520165D
	SHA256	34DF739526C114BB89470B3B650946CBF7335CB4A2206489534FB05C1FC143A8
listdlls.exe	MD5	5245F11D3664BB6C5956E58C83BB8C5F
	SHA1	CF1D18CF4EE232052DFD7F1A6100E86D804E1B0B
	SHA256	020D4B225126F93254A15DAE24C80C0B889D945F7A3E552E3B0F2B35939A8D2B
logonsessions.exe	MD5	68767E20FD31D5348F5979C00AFE4F7F
	SHA1	7C762173D3C7F4366371E2A475B8B5BAFB5BF64F
	SHA256	4296ECB7BD7BAF0BFEB364A88B4C87695E5F16F193E5AC954E8A51BD5D58DB54
nbtstat.exe	MD5	D6A9FE571146099D6D75A8E4E7871506
	SHA1	68DBA140959ED155F720060C5466F5FD90A176F6
	SHA256	F63D1A87E8D264321BD2EF30B017758EF77CF741849F3F7F214BB169C0C9A461
net.exe	MD5	63DD6FBAABF881385899FD39DF13DCE3

	SHA1	B25697B250631BB09D27E259A2D280CFA97CA456
	SHA256	3B9AD8E2C1D03FF941A7C9192A605F31671B107DEF6FF503A71A0FB2C5BBD659
netstat.exe	MD5	6F39F6F48CD4828B2C87EB2D2CAB45A5
	SHA1	AC4A74D027962554608CE9A90BB8204788ABCD3C
	SHA256	5C748735F5D876A84163D16B042F3AC92D27131B352012E42E16FAE89D1A890D
pslist.exe	MD5	AD06AA36E330434560593590330222E6
	SHA1	4273B7BD38FC1F203CCC5FDFA1F7331B2683F001
	SHA256	09174BF3DC391920CC89760D3D1933A0D41E573111897B0EB3C8472758FDDBE5
psloggedon.exe	MD5	08DADAC8C7A951CBEC90C10026BA74B3
	SHA1	A9B37AF96190ADFCF36FB6301B1E07DA1C5C4443
	SHA256	40C2D8D7E58DC4E0AF897A6CF6E662A6BE914C93D5EC5B6DB570E5F4855E4E78
reg.exe	MD5	9D0B3066FE3D1FD345E86BC7BCCED9E4
	SHA1	E05984A6671FCFECBC465E613D72D42BDA35FD90
	SHA256	4E66B857B7010DB8D4E4E28D73EB81A99BD6915350BB9A63CD86671051B22F0E
sc.exe	MD5	4EBBC2B0AD7F9075AE9D6835D2A62B6E
	SHA1	DB1F81F5E209FED6DF3255F6C820555CF17A839C
	SHA256	EAAB690EBD8DDF9AE452DE1BC03B73C8154264DBD7A292334733B47A668EBF31
tasklist.exe	MD5	A9A00E71E3DD67B029FC904FE3BB61DA
	SHA1	430AA43010EEF3CD43ED445777F3D5CCF6BC4C27
	SHA256	AD3E811249DA750D80F2762C3AEB403780C1B69D05911E3C9950A7DAED9E6670

2. Tool Deployment

With the toolset assembled, and a means for verifying its integrity in place, we next focused on a method for employing the tools in an automated manner. The integral part of making this work was the Sysinternals tool psexec. This tool allows for remote connections to computers in a Windows environment as well as the remote execution of programs. Incorporating this tool we constructed two batch files. The first batch file, which we named connect.bat, remotely connects to all of the computers in the domain using psexec and then copies the second batch file to those computers and executes it. The second batch file, which we named toolscript.bat, reaches back to the triage machine and executes the tools from it; it then redirects all of the output from the tools to a separate file for each machine in the results folder on the triage machine. The contents of the two batch files can be found in Appendix C.

We then ran our toolset across the entire enterprise. The results obtained served as our baseline. After the insertion of malware into the network, which is explained in the next chapter, we ran the toolset again. We then analyzed the two results sets for discrepancies that can be used to signify a compromise in the network.

V. MALWARE IMPLEMENTATION

With the scenario environment built, and the toolkit developed, the next step was to test the toolkit. In order to test the efficacy of the toolkit in identifying and classifying compromises in a network, we had to create a compromise. To do this we infected one of the machines with malware. When trying to determine the right malware to use in this project we had several objectives in mind. First we wanted to use a well-developed software package. We did not want to have to troubleshoot unknown issues while trying to deploy it. Next, we wanted to implement software that had an intuitive user interface not requiring an in-depth knowledge of the code in order to use it. Finally, we wanted to use malware that has been used in notable attacks and has actual pertinence to present day computer security. With these objectives in mind, we chose the Remote Access Tool (RAT), Poison Ivy.

A. BACKGROUND

Poison Ivy is a backdoor program that has been actively recognized on the Internet since 2005. Poison Ivy has been at the root of many notable corporate network compromises. The most recognized being the attack on RSA that compromised a large number of their SecureID tokens in 2011. Attackers have also notably used Poison Ivy in concerted attacks against chemical makers, government agencies, defense contractors, and human rights groups [41]. The tool itself is not very sophisticated, but it does provide a persistent and often well hidden connection to compromised machines inside a network. This gives attackers the ability to perform more sophisticated attacks from within the target network.

B. CAPABILITIES

The list below from Trend Micro’s analysis of Poison Ivy details many of its capabilities [43].

Capture screen, audio, and webcam

List active ports

Log keystrokes

Manage open windows

Manage passwords

Manage registry, processes, services, devices, and installed applications

Perform multiple simultaneous transfers of files

Remote shell access

Relay server

Search files

Update, restart, and terminate itself

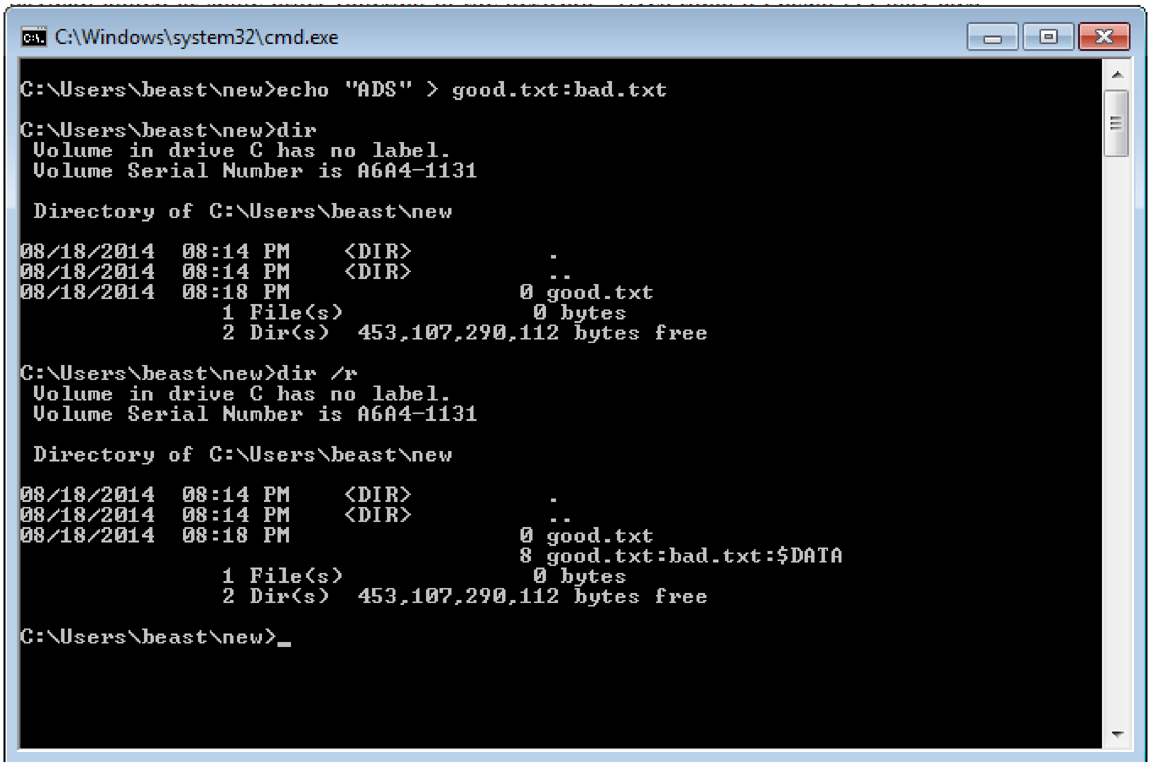
Many of these capabilities are well-suited for a program used in the exfiltration of data in cyber-espionage attacks. However, some of the capabilities lend themselves to more sophisticated attacks. First is the ability to manage passwords. This lets the attackers potentially escalate their privilege level on the now compromised network, gaining access to new areas and more information. Also, there is the remote shell capability. With this, an attacker can use Poison Ivy to deliver an even more nefarious payload targeting other devices and services in the network. Even though Poison Ivy may lack the refinement of current malware it still has the capability to wreak havoc upon a network.

C. COMMAND AND CONTROL

The nomenclature surrounding Poison Ivy can be somewhat misleading. It consists of two components, a client and a server; however, the client is software on the machine that the attacker is using, while the server is the software that is embedded in the target network. It communicates between the client and server using a custom protocol over TCP. It can be configured to use any port number but the default in the application is port 3460. It uses the Camelia cipher with a 256 bit key for encryption and Microsoft’s LZNT1 algorithm for compression [42]. For the attacker, the command and control software provides a very intuitive graphical user interface, which provides point and click access to all of the aforementioned capabilities.

D. CLIENT INFECTION

The primary way Poison Ivy infects new machines is from downloads on malicious websites. The file size for the tool is less than 10 kilobytes (kB) before the addition of any wrappers or obfuscation [42]. Once executed, the file copies itself to a location predefined by the attacker and inserts registry entries for persistence. Next, Poison Ivy injects itself into the process of the default browser for the machine, further hiding its operations from discovery. It does this by starting the browser process with the *–nohome* argument which starts the process but does not open a Window for the browser. Finally, it can use Alternate Data Streams (ADS), an obscure component of the New Technology File System (NTFS), to hide itself within other files [44]. A normal directory listing will not display the ADS the dir –R command or a tool like Sysinternal's sigcheck must be used. Figure 6 gives an example of this.



```
C:\Users\beast\new>echo "ADS" > good.txt:bad.txt

C:\Users\beast\new>dir
 Volume in drive C has no label.
 Volume Serial Number is A6A4-1131

 Directory of C:\Users\beast\new

08/18/2014  08:14 PM    <DIR>          .
08/18/2014  08:14 PM    <DIR>          ..
08/18/2014  08:18 PM                 0 good.txt
               1 File(s)              0 bytes
               2 Dir(s)  453,107,290,112 bytes free

C:\Users\beast\new>dir /r
 Volume in drive C has no label.
 Volume Serial Number is A6A4-1131

 Directory of C:\Users\beast\new

08/18/2014  08:14 PM    <DIR>          .
08/18/2014  08:14 PM    <DIR>          ..
08/18/2014  08:18 PM                 0 good.txt
                                     8 good.txt:bad.txt:$DATA
               1 File(s)              0 bytes
               2 Dir(s)  453,107,290,112 bytes free

C:\Users\beast\new>_
```

Figure 6. Hiding files in alternate data streams

E. IMPLEMENTATION

To insert Poison Ivy in our lab environment we installed a new virtual machine with the Kali Linux operating system. Kali is a distribution dedicated to penetration testing with over 300 tools for testing network security [45]. We placed this machine outside of the enterprise network in the 192.168.0.0/24 network. It has a manually configured IP address of 192.168.0.10. Once the Kali machine was fully installed we then downloaded and installed Poison Ivy. Instructions for the installation and configuration of Poison Ivy can be found in Appendix D. We followed these instructions to create the client interface on our Kali machine, and to create the server.exe file which served as the RAT infecting a machine on the enterprise network. Finally, we started the Apache web server service on the Kali machine to serve as our vector of attack. An unsuspecting user that was logged into the ALPHAWS01 workstation browsed to the http://192.168.0.10/share website and clicked the link. This executed the server.exe file on the ALPHAWS01 machine, installing Poison Ivy and providing administrative access to this machine from the Kali machine. This not only allowed the attacker on the Kali machine access to the ALPHAWS01 machine, but to all of the resources available on the alpha.aco domain as well as the other domains on the enterprise network due to the trust relationships.

VI. ANALYSIS AND RECOVERY

At this point in the project a computer on the network had been compromised by Poison Ivy. We had two sets of outputs from our developed native toolset: one to serve as the baseline for known good operations and the other from a compromised machine. This chapter presents the analysis of these two outputs for the purpose of discovering indicators of compromise. These IOCs are specific to Poison Ivy. Had we used another form of malware, we would have discovered different indicators of compromise. We examined the output of each tool and described possible IOCs they could provide. Outputs that did provide evidence of a compromise are shown in this chapter. The other outputs that did not are presented in Appendix F.

Once we analyzed the outputs and determined that we had a compromise on the system the focus shifted to recovery. The second section of this chapter focuses on this. It describes the actions that were taken to recover network functionality through the use of the mirrored VLAN. It then covers some of the other capabilities use off the mirrored VLAN facilitated.

A. TOOL OUTPUT ANALYSIS

The following section provides the analysis of each tool's output in the order in which they were run from the batch file.

1. Executables

The psloggedon command output displays the users that are logged onto the system. As a remote triage tool, it can indicate unauthorized access if it displays users that should not be on the system. In our analysis there were no discrepancies in the output from the two scans. Similar to psloggedon, logonsessions provides the usernames of all users logged onto a system. In addition, it also displays their logon type, the authentication mechanism used, and the processes that are running in the session. Unexplained logons and even legitimate logons that are usually not active during the time of the survey are typical indications of compromise. Suspect and unknown processes in this output are also considered possible IOCs. Moreover, the output from logonsessions can be used in conjunction with the output from other tools such as handle to further investigate suspicious activity. If a process listed in handle's output is running for a user that has no entry in logonsessions it warrants further investigation. "Backdoors and Trojans such as the infamous SubSeven allow users to log in to the Trojan via a raw Transmission Control Protocol connection, bypassing the windows authentication mechanisms," and not creating an entry in logonsessions [46, p. 19]. Table 10 provides a comparison of the logonsessions output from the baseline and the compromised machine. In the infected machine's output, process id 3652 iexplore.exe appears to be a legitimate process running in the administrator's session. This process is later revealed to be malicious; however, at this stage it had not been identified as an indicator of compromise. In the tables that follow in this chapter, yellow highlighted items are discovered IOCs and blue highlighted items are compromised processes that were not evident from the tool output.

Table 10. Output of the logonsessions –p command

Baseline Output	Infected Output
Logonsesions v1.21 Copyright (C) 2004–2010 Bryce Cogswell and Mark Russinovich Sysinternals - wwww.sysinternals.com [6] Logon session 00000000:001f429b: User name: ALPHA\Administrator Auth package: Kerberos Logon type: Interactive Session: 1 Sid: S-1-5-21-3079887268-1858392370-3246419219-500 Logon time: 8/18/2014 3:56:56 PM Logon server: ALPHADC01 DNS Domain: ALPHA.ACO UPN: Administrator@alpha.aco 868: taskhost.exe 1296: dwm.exe 1504: explorer.exe 1540: cmd.exe 1108: conhost.exe	Logonsesions v1.21 Copyright (C) 2004–2010 Bryce Cogswell and Mark Russinovich Sysinternals - wwww.sysinternals.com [5] Logon session 00000000:0001a854: User name: ALPHA\Administrator Auth package: Kerberos Logon type: Interactive Session: 1 Sid: S-1-5-21-3079887268-1858392370-3246419219-500 Logon time: 8/22/2014 12:38:25 PM Logon server: ALPHADC01 DNS Domain: ALPHA.ACO UPN: Administrator@alpha.aco 1416: taskhost.exe 1672: dwm.exe 1816: explorer.exe 3652: iexplore.exe

The next tool run in the survey was net session. The net session command can display the usernames of accounts that are remotely accessing the system. It also displays the client type from which they are connecting and any shared resources, such as drives, they are connected to on this system. Observing unauthorized users or multiple instances of the same account logged on both locally and remotely to a system could also be an IOC. The analysis we performed on these outputs presented no IOCs.

Tasklist outputs a list of running processes, the process identification, session name, number, and the memory usage. The analyst's familiarity with processes should allow for rogue or unknown processes to register as being unfamiliar. Unfamiliar processes can be researched on repository sites such ProcessLibrary.com and TaskList.org. In addition, Appendix B of this document contains many of the common processes in Windows operating systems and their functions within the OS. An investigator will always want to know what processes are running on a potentially compromised system, as they could provide correlation of compromises on multiple systems. Additional switches such as "tasklist /SVC" show service name to process relationship and "tasklist /M" shows the dynamic linked libraries (DLL) associated with the process [46, pp. 23–26]. The infected host output contains malicious iexplore.exe PID 3652; however, at this stage of the survey, it appears normal and not an IOC.

The next tool, pslist –t, enhances the output from tasklist. It displays parent and child process relationships in a tree format. The analyst should be familiar with the order in which processes are started during system startup and use the displayed CPU time and elapsed time to provide insight into the process's correct elapsed time with respect to when the process starts during system startup. For instance, system startup processes such as lsass.exe, csrss.exe, smss.exe should be in the beginning of the list, not children of the explorer process. Processes with odd child parent relationships should be investigated as a potential IOC [46, pp. 23–26]. Table 11 shows the normal baseline output and infected output which contains the malicious process id 3652, iexplore, which is not flagged as an IOC at this stage of the survey.

Table 11. Output of pslist –t command

Baseline Output

```
Process information for ALPHAWS01:

Name                    Pid Pri Thd  Hnd      VM      WS    Priv
Idle                      0   0   1    0       0      24       0
  System                  4   8  87  466    2560    1032      48
    smss                252  11   2   29    4048     532     220
csrss                   336  13   9  337   35304    2520    1164
  conhost              1188   8   2   35   21684    2204     552
wininit                 384  13   3   74   34208    2564     872
  services              480   9   6  196   31880    4960    3508
    sppsvc              312   8   4  146   30136    5568    2012
    svchost             600   8   9  334   36724    5004    2396
    svchost             680   8   7  222   27176    4348    2256
    svchost             804   8  18  437   53500    8948    8368
    svchost             840   8  14  375   95924   26984   28728
      dwm              1296   8   3   69   41156    3072    1000
    taskhost            868   8   8  160   44396    5044    2432
    svchost             880   8  29  924   98568   12124   13448
    svchost             988   8  13  351   51040    7080    4900
    svchost            1084   8  17  485   73056   10756   13460
    spoolsv            1172   8  12  263   59328    5180    4160
    svchost            1208   8  18  303   40348    6424    7544
    svchost            1332   8   9  150   36864    4768    3212
    SearchIndexer      1428   8  14  676  107304   11684   26068
    svchost            1752   8   5   96   26680    2920    1108
    svchost            1872   8   8  302   47480    3672    2052
    PSEXESVC           3172   8   9  116   40508    4140    1624
      cmd              2040   8   1   32   21200    2572    1832
        pslist         2872  13   1  138   41436    4100    1572
  lsass                 488   9   9  655   34804    7024    3296
  lsm                   496   8   9  135   14392    2284    1044
csrss                   396  13   8  178   37244    3964    1228
  conhost              1108   8   2   51   47608    3968     908
winlogon                436  13   3  108   40500    3224    1564
explorer               1504   8  18  656  176400   29100   16252
  cmd                  1540   8   1   22   31528    2144    1732
  iexplore             3788   8  12  390  136968   19524    7728
    iexplore           3872   8  14  352  113496   18160    5144
```

Infected Output

```
Process information for ALPHAWS01:

Name                    Pid Pri Thd  Hnd      VM      WS    Priv
Idle                      0   0   1    0       0      24       0
  System                  4   8  81  496    2176     636      44
    smss                252  11   2   29    4048     536     216
csrss                   336  13   8  350   35112    2556    1132
  conhost              3960   8   2   33   21620    2208     548
wininit                 384  13   3   72   34208    2884     944
  services              480   9   6  189   28836    5164    3344
    svchost             604   8   9  336   35868    5420    2516
      WmiPrvSE         2884   8   6  109   27888    4472    1732
    svchost             668   8   7  218   27176    4396    2044
    svchost             716   8  18  440   53500    9460    8404
    svchost             840   8  16  387   97168   32140   27944
      dwm              1672   8   3   68   41156    3644    1000
    svchost             880   8  33  922   93572   19852   12772
    SearchIndexer       972   8  12  634   82964    8528   15344
    svchost             988   8  16  386   52744    8188    4840
    svchost            1080   8  16  374   59832    9064   10960
    spoolsv            1172   8  12  264   59328    7692    4164
    svchost            1208   8  18  299   45628    6464    7876
    svchost            1332   8  12  208   38024    5136    3272
    taskhost           1416   8   8  156   44396    5324    2452
    svchost            1756   8   5   94   26680    3468    1104
    wmpnetwk           1976   8   9  219   77904    2472    3244
    svchost            1984   8   9  310   47736    5156    2072
    sppsvc             2036   8   4  146   30136    6760    1936
    PSEXESVC           3852   8   9  114   40508    4292    1720
      cmd              3952   8   1   30   22224    2744    1980
  lsass                 492   9   7  627   34292    7532    2900
  lsm                   500   8  10  136   14648    2480    1088
csrss                   396  13   8  318   37116    4424    1276
  conhost              3556   8   2   53   47672    3980     880
winlogon                436  13   3  108   40500    3932    1420
explorer               1816   8  26  821  224480   45420   33296
  iexplore             3652   8   4  109   54984    4368    1772
  cmd                  3964   8   1   24   28904    2448    1852
    pslist             4076  13   1  130   53212    4104    1608
```

Listdlls is useful for verifying known processes and gleaning information on unknown or rogue processes. The observation of the command line arguments, process, process id, Dynamic Linked Library (DLL) and path of the DLL can provide strong evidence of abnormal behavior. In the output below the highlighted entry is an indicator of compromise. There are half as many DLLs being called by process 3652 vice process 3872 of iexeplorer.exe. This is unusual and further investigation reveals the –nohome switch. This allows iexplore.exe to run in the background without opening a browser page. This process is actually Poison Ivy which has injected itself into iexplorer.exe. This IOC would still be visible if time stamp modification or executable renaming obfuscation techniques were in use. Table 12 displays the outputs of this command.

Table 12. Output of listdlls command

Baseline Output	Infected Output
ListDLLs v3.1 - List loaded DLLs Copyright (C) 1997–2011 Mark Russinovich Sysinternals - www.sysinternals.com -- iexplore.exe pid: 3872 Command line: “C:\Program Files\Internet Explorer\iexplore.exe” SCODEF:3788 CREDAT:14337 Base Size Path 0x008d0000 0xa6000 C:\Program Files\Internet Explorer\iexplore.exe 0x77680000 0x13c000 C:\Windows\SYSTEM32\ntdll.dll 0x77140000 0xd4000 C:\Windows\system32\kernel32.dll 0x75850000 0x4a000 C:\Windows\system32\KERNELBASE.dll 0x77220000 0xa0000 C:\Windows\system32\ADVAPI32.dll 0x77060000 0xac000 C:\Windows\system32\msvcrt.dll 0x75df0000 0x19000 C:\Windows\SYSTEM32\sechost.dll 0x75bd0000 0xa1000 C:\Windows\system32\RPCRT4.dll 0x773b0000 0xc9000 C:\Windows\system32\USER32.dll 0x76dc0000 0x4e000 C:\Windows\system32\GDI32.dll 0x76bb0000 0xa000 C:\Windows\system32\LPK.dll 0x75c80000 0x9d000 C:\Windows\system32\USP10.dll 0x77000000 0x57000 C:\Windows\system32\SHLWAPI.dll 0x75e20000 0xc49000 C:\Windows\system32\SHELL32.dll 0x76bc0000 0x15c000 C:\Windows\system32\ole32.dll 0x77480000 0x1f9000 C:\Windows\system32\iertutil.dll 0x76a70000 0x135000 C:\Windows\system32\urlmon.dll 0x772c0000 0x8f000 C:\Windows\system32\OLEAUT32.dll 0x758c0000 0x11c000 C:\Windows\system32\CRYPT32.dll 0x75840000 0xc000 C:\Windows\system32\MSASN1.dll 0x76d20000 0x1f000 C:\Windows\system32\IMM32.DLL 0x75d20000 0xcc000 C:\Windows\system32\MSCTF.dll 0x6ed80000 0xa7c000 C:\Windows\system32\IEFRAME.dll 0x77810000 0x5000 C:\Windows\system32\PSAPI.DLL 0x6ed40000 0x3c000 C:\Windows\system32\OLEACC.dll 0x74860000 0x19e000 C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc\comctl32.dll 0x76d40000 0x7b000 C:\Windows\system32\comdlg32.dll 0x6e290000 0x35000 C:\Program Files\Internet Explorer\IEShims.dll 0x75720000 0xc000 C:\Windows\system32\CRYPTBASE.dll 0x74820000 0x40000 C:\Windows\system32\uxtheme.dll 0x757c0000 0xe000 C:\Windows\system32\RpcRtRemote.dll 0x743f0000 0x13000 C:\Windows\system32\dwmapi.dll 0x77820000 0x83000 C:\Windows\system32\CLBCatQ.DLL 0x74410000 0xf5000 C:\Windows\system32\propsys.dll 0x757d0000 0xb000 C:\Windows\system32\profapi.dll 0x76e60000 0x19d000 C:\Windows\system32\SETUPAPI.dll 0x75aa0000 0x27000 C:\Windows\system32\CFGMGR32.dll 0x758a0000 0x12000 C:\Windows\system32\DEVOBJ.dll 0x75250000 0x16000 C:\Windows\system32\CRYPTSP.dll 0x74ff0000 0x3b000 C:\Windows\system32\rsaenh.dll 0x6e3b0000 0x2b000 C:\Program Files\Internet Explorer\ieproxy.dll 0x75ad0000 0xf4000 C:\Windows\system32\WININET.dll 0x75e10000 0x3000 C:\Windows\system32\Normaliz.dll 0x756b0000 0x1a000 C:\Windows\system32\SspiCli.dll 0x777d0000 0x35000 C:\Windows\system32\ws2_32.DLL 0x777c0000 0x6000 C:\Windows\system32\NSI.dll	ListDLLs v3.1 - List loaded DLLs Copyright (C) 1997–2011 Mark Russinovich Sysinternals - www.sysinternals.com -- ---- iexplore.exe pid: 3652 Command line: “C:\Program Files\Internet Explorer\iexplore.exe” -nohome Base Size Path 0x01310000 0xa6000 C:\Program Files\Internet Explorer\iexplore.exe 0x77a10000 0x13c000 C:\Windows\SYSTEM32\ntdll.dll 0x77760000 0xd4000 C:\Windows\system32\kernel32.dll 0x75be0000 0x4a000 C:\Windows\system32\KERNELBASE.dll 0x76360000 0xa0000 C:\Windows\system32\ADVAPI32.dll 0x76400000 0xac000 C:\Windows\system32\msvcrt.dll 0x760b0000 0x19000 C:\Windows\SYSTEM32\sechost.dll 0x774c0000 0xa1000 C:\Windows\system32\RPCRT4.dll 0x77b70000 0xc9000 C:\Windows\system32\USER32.dll 0x779c0000 0x4e000 C:\Windows\system32\GDI32.dll 0x77b50000 0xa000 C:\Windows\system32\LPK.dll 0x75e60000 0x9d000 C:\Windows\system32\USP10.dll 0x77670000 0x57000 C:\Windows\system32\SHLWAPI.dll 0x76670000 0xc49000 C:\Windows\system32\SHELL32.dll 0x77860000 0x15c000 C:\Windows\system32\ole32.dll 0x772c0000 0x1f9000 C:\Windows\system32\iertutil.dll 0x76530000 0x135000 C:\Windows\system32\urlmon.dll 0x75f10000 0x8f000 C:\Windows\system32\OLEAUT32.dll 0x75d40000 0x11c000 C:\Windows\system32\CRYPT32.dll 0x75bd0000 0xc000 C:\Windows\system32\MSASN1.dll 0x77840000 0x1f000 C:\Windows\system32\IMM32.DLL 0x75fe0000 0xcc000 C:\Windows\system32\MSCTF.dll 0x75fa0000 0x35000 C:\Windows\system32\ws2_32.DLL 0x77b60000 0x6000 C:\Windows\system32\NSI.dll 0x75610000 0x3c000 C:\Windows\system32\mswsock.dll 0x750e0000 0x5000 C:\Windows\System32\wshtcpip.dll 0x74ee0000 0x12000 C:\Windows\system32\mpr.dll 0x6fa20000 0x13000 C:\Windows\system32\avicap32.dll 0x74f00000 0x32000 C:\Windows\system32\WINMM.dll 0x75050000 0x9000 C:\Windows\system32\VERSION.dll 0x6f9f0000 0x21000 C:\Windows\system32\MSVFW32.dll 0x75c60000 0x84000 C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_5.82.7600.16385_none_ebf82fc36c758ad5\COMCTL32.dll 0x6fdb0000 0xd000 C:\Windows\system32\pstorec.dll 0x73ef0000 0x14000 C:\Windows\system32\ATL.DLL 0x73e30000 0x1c000 C:\Windows\system32\iphlpapi.dll 0x73e20000 0x7000 C:\Windows\system32\WINNSI.DLL 0x75a40000 0x1a000 C:\Windows\system32\SspiCli.dll 0x74790000 0x39000 C:\Windows\system32\MMDevAPI.DLL 0x74690000 0xf5000 C:\Windows\system32\PROPSYS.dll

0x750d0000 0x44000 C:\Windows\system32\dnsapi.DLL 0x73aa0000 0x1c000 C:\Windows\system32\iphlpapi.DLL 0x73a90000 0x7000 C:\Windows\system32\WINNSI.DLL 0x756d0000 0x4b000 C:\Windows\system32\apphelp.dll 0x736f0000 0x21000 C:\Windows\system32\ntmarta.dll 0x76e10000 0x45000 C:\Windows\system32\WLDAP32.dll 0x74cc0000 0x9000 C:\Windows\system32\VERSION.dll 0x74070000 0x52000 C:\Windows\system32\RASAPI32.dll 0x74050000 0x15000 C:\Windows\system32\rasman.dll 0x74040000 0xd000 C:\Windows\system32\rtutils.dll 0x702f0000 0x6000 C:\Windows\system32\sensapi.dll 0x75210000 0x3c000 C:\Windows\system32\mswsock.dll 0x74d50000 0x5000 C:\Windows\System32\wshtcpip.dll 0x73c20000 0x10000 C:\Windows\system32\NLAapi.dll 0x73580000 0x6000 C:\Windows\system32\rasadhlp.dll 0x6e2e0000 0x2e000 C:\Windows\system32\MLANG.dll 0x73070000 0x8000 C:\Windows\System32\winrnr.dll 0x73060000 0x10000 C:\Windows\system32\napinsp.dll 0x73020000 0x12000 C:\Windows\system32\pnrpnsp.dll 0x75200000 0x6000 C:\Windows\System32\wship6.dll 0x73930000 0x38000 C:\Windows\System32\fwpuclnt.dll 0x75730000 0x5f000 C:\Windows\system32\SXS.DLL 0x6d5e0000 0x5b2000 C:\Windows\System32\mshtml.dll 0x70640000 0x2a000 C:\Windows\System32\msls31.dll 0x6e260000 0x30000 C:\Windows\System32\iepeers.dll 0x71d30000 0x51000 C:\Windows\System32\WINSPOOL.DRV 0x702d0000 0xb000 C:\Windows\system32\msimtf.dll 0x6e1a0000 0xb2000 C:\Windows\System32\jscript.dll 0x702c0000 0xb000 C:\Windows\system32\ImgUtil.dll 0x70280000 0xe000 C:\Windows\System32\pngfilt.dll 0x702e0000 0x5000 C:\Windows\system32\msimg32.dll 0x73e10000 0x32000 C:\Windows\system32\WINMM.dll 0x74510000 0x39000 C:\Windows\system32\MMDevAPI.DLL	

The handle command outputs all running processes and their relative executable path. This enables the analyst to determine the file path location(s) of a process in addition to its active file descriptors. When an unknown process is running, handle can be used to provide an analyst with further clues about its origin and behaviors. In the handle.exe output below there is a difference between the standard operation of iexplore.exe in a clean machine and the Poison Ivy usage of iexplore.exe in a tainted machine. The iexplorer.exe process in the infected output was started by the server.exe file in the system 32 directory. Which is atypical since the iexeplorer.exe file resides in the “Program Files” directory. Table 13 presents the outputs of this command.

Table 13. Output of handle command

Baseline Output	Infected Output
Handle v3.51 Copyright (C) 1997–2013 Mark Russinovich Sysinternals - www.sysinternals.com iexplore.exe pid: 3872 ALPHA\administrator 8: File (RW-) C:\Users\administrator\Desktop 38: File (R-D) C:\Program Files\Internet Explorer\en-US\iexplore.exe.mui C0: File (RW-) C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc D0: Section \Sessions\1\BaseNamedObjects\Internet Explorer Immutable Application State (00000ECC-0000-0000-0000-000000000000) EC: Section \Sessions\1\BaseNamedObjects\windows_shell_global_counters F0: File (RW-) C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc 110: Section \Sessions\1\BaseNamedObjects\ie_lcie_main_ecc 114: Section \Sessions\1\BaseNamedObjects\Isolation Process Registry (427E9554-2732-11E4-8C8C-0050569C73C3) 118: Section \Sessions\1\BaseNamedObjects\Isolation Signal Registry (427E9554-2732-11E4-8C8C-0050569C73C3, 0) 18C: Section \BaseNamedObjects__ComCatalogCache__ 194: Section \BaseNamedObjects__ComCatalogCache__ 1A0: Section \BaseNamedObjects\windows_shell_global_counters 1B8: Section \Sessions\1\BaseNamedObjects\IEFrame!GetAsyncKeyStateSharedMem!3788 204: Section \BaseNamedObjects\C:*ProgramData*Microsoft*Windows*Caches*cversions.2.ro 220: Section \BaseNamedObjects\C:*ProgramData*Microsoft*Windows*Caches*cversions.2.ro 224: Section \Sessions\1\BaseNamedObjects\windows_shell_global_counters 24C: File (RW-) C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc	Handle v3.51 Copyright (C) 1997–2013 Mark Russinovich Sysinternals - www.sysinternals.com iexplore.exe pid: 3652 ALPHA\administrator 8: File (RW-) C:\Windows\System32 38: File (R-D) C:\Program Files\Internet Explorer\en-US\iexplore.exe.mui AC: File (RW-) C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_5.82.7600.16385_none_ebf82fc36c758ad5 BC: File (R-D) C:\Windows\System32\en-US\msvfw32.dll.mui C0: File (R-D) C:\Windows\System32\en-US\avicap32.dll.mui 138: Section \BaseNamedObjects\mmGlobalPnpInfo C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc 360: File (---) C:\Windows\System32\server.exe

The ipconfig command outputs network configuration information critical to identifying the system under investigation. Any changes to the IP or MAC addresses of the computer should be investigated further as a potential IOC. Special attention must be paid to computers with dual NICs. Systems with dual NICs are high value targets as they allow an attacker to pivot from compromised network enclaves into deeper out of band enclaves of the enterprise network. There were no discrepancies between the outputs we analyzed.

Following in the networking aspects of the system, the next tool in the survey was netstat. The netstat –an command provides the active network connections on the system. Being familiar with the listening ports and their associated services on a baseline machine allows an analyst to notice suspect network connections. Also depending on the purpose of the system, established connections should be looked for. In the table below, the highlighted entry on port 3460 is an indicator of Poison Ivy. Table 14 compares the baseline and infected outputs of this command.

Table 14. Output of netstat –an command

Baseline Output	Infected Output
Active Connections Proto Local Address Foreign Address State TCP 0.0.0.0:135 0.0.0.0:0 LISTENING TCP 0.0.0.0:445 0.0.0.0:0 LISTENING TCP 0.0.0.0:5357 0.0.0.0:0 LISTENING TCP 0.0.0.0:49152 0.0.0.0:0 LISTENING TCP 0.0.0.0:49153 0.0.0.0:0 LISTENING TCP 0.0.0.0:49154 0.0.0.0:0 LISTENING TCP 0.0.0.0:49172 0.0.0.0:0 LISTENING TCP 0.0.0.0:49173 0.0.0.0:0 LISTENING TCP 0.0.0.0:49184 0.0.0.0:0 LISTENING TCP 10.1.0.25:139 0.0.0.0:0 LISTENING TCP 10.1.0.25:445 10.1.0.2:56817 ESTABLISHED TCP 10.1.0.25:51273 10.1.0.2:445 ESTABLISHED TCP 10.1.0.25:51277 10.1.0.2:445 ESTABLISHED TCP [::]:135 [::]:0 LISTENING TCP [::]:445 [::]:0 LISTENING TCP [::]:5357 [::]:0 LISTENING TCP [::]:49152 [::]:0 LISTENING TCP [::]:49153 [::]:0 LISTENING TCP [::]:49154 [::]:0 LISTENING TCP [::]:49172 [::]:0 LISTENING TCP [::]:49173 [::]:0 LISTENING TCP [::]:49184 [::]:0 LISTENING New connections will not be remembered. There are no entries in the list.	Active Connections Proto Local Address Foreign Address State TCP 0.0.0.0:135 0.0.0.0:0 LISTENING TCP 0.0.0.0:445 0.0.0.0:0 LISTENING TCP 0.0.0.0:5357 0.0.0.0:0 LISTENING TCP 0.0.0.0:49152 0.0.0.0:0 LISTENING TCP 0.0.0.0:49153 0.0.0.0:0 LISTENING TCP 0.0.0.0:49154 0.0.0.0:0 LISTENING TCP 0.0.0.0:49155 0.0.0.0:0 LISTENING TCP 0.0.0.0:49156 0.0.0.0:0 LISTENING TCP 0.0.0.0:49157 0.0.0.0:0 LISTENING TCP 10.1.0.25:139 0.0.0.0:0 LISTENING TCP 10.1.0.25:445 10.1.0.2:57947 ESTABLISHED TCP 10.1.0.25:49185 192.168.1.10:**3460** ESTABLISHED TCP 10.1.0.25:49503 10.1.0.2:445 ESTABLISHED TCP 10.1.0.25:49511 10.1.0.2:445 ESTABLISHED TCP [::]:135 [::]:0 LISTENING TCP [::]:445 [::]:0 LISTENING TCP [::]:5357 [::]:0 LISTENING TCP [::]:49152 [::]:0 LISTENING TCP [::]:49153 [::]:0 LISTENING TCP [::]:49154 [::]:0 LISTENING TCP [::]:49155 [::]:0 LISTENING TCP [::]:49156 [::]:0 LISTENING TCP [::]:49157 [::]:0 LISTENING New connections will not be remembered. There are no entries in the list.

Observing the connection to an unknown or suspect port should be followed up by running netstat –anob. This displays the relationship between the executable, the process identifier, and the associated established network connection. [46, p. 31] It appears the iexplore.exe PID 3652 has established a connection on port 3460. Usually iexplore.exe is a legitimate binary; however, an established connection to a port other than 80 or 443 is abnormal. In this instance Poison Ivy used process injection to establish an outbound connection. Table 15 shows the output of the netstat -anob command with the iexplorer.exe program as the owner of the connection. A seasoned intruder could change this to port 443 making this appear like encrypted web traffic and making it more difficult to identify the activity as an IOC. Referencing the output of netstat -anob against those from handle and listdlls can help when examining browser sessions that appear to be legitimate usage.

Table 15. Output of the netstat –anob command

```
Active Connections

  Proto  Local Address      Foreign Address     State        PID
  TCP    0.0.0.0:135        0.0.0.0:0           LISTENING    668     RpcSs    [svchost.exe]
  TCP    0.0.0.0:445        0.0.0.0:0           LISTENING    4       Can   not   obtain   ownership
information
  TCP    0.0.0.0:5357       0.0.0.0:0           LISTENING    4        Can   not   obtain   ownership
information
  TCP    0.0.0.0:49152      0.0.0.0:0           LISTENING    384     [wininit.exe]
  TCP    0.0.0.0:49153      0.0.0.0:0           LISTENING    716     eventlog  [svchost.exe]
  TCP    0.0.0.0:49154      0.0.0.0:0           LISTENING    880     Schedule  [svchost.exe]
  TCP    0.0.0.0:49155      0.0.0.0:0           LISTENING    480     [services.exe]
  TCP    0.0.0.0:49156      0.0.0.0:0           LISTENING    1756    PolicyAgent        [svchost.exe]
  TCP    0.0.0.0:49157      0.0.0.0:0           LISTENING    492     [lsass.exe]
  TCP    10.1.0.25:135      10.1.0.50:49229     ESTABLISHED  668     RpcSs     [svchost.exe]
  TCP    10.1.0.25:139      0.0.0.0:0           LISTENING    4       Can   not   obtain   ownership
information
  TCP    10.1.0.25:445      10.1.0.50:49225     ESTABLISHED  4       Can   not   obtain   ownership
information
  TCP    10.1.0.25:49155    10.1.0.50:49230     ESTABLISHED  480     [services.exe]
  TCP    10.1.0.25:49185    192.168.1.10:3460   ESTABLISHED  3652    [iexplore.exe]
  TCP    10.1.0.25:49809    10.1.0.50:445       ESTABLISHED  4       Can   not   obtain   ownership
information
  TCP    [::]:135           [::]:0              LISTENING    668     RpcSs    [svchost.exe]
  TCP    [::]:445           [::]:0              LISTENING    4       Can   not   obtain   ownership
information
  TCP    [::]:5357          [::]:0              LISTENING    4        Can   not   obtain   ownership
information
  TCP    [::]:49152         [::]:0              LISTENING    384     [wininit.exe]
  TCP    [::]:49153         [::]:0              LISTENING    716     eventlog  [svchost.exe]
  TCP    [::]:49154         [::]:0              LISTENING    880     Schedule  [svchost.exe]
  TCP    [::]:49155         [::]:0              LISTENING    480     [services.exe]
  TCP    [::]:49156         [::]:0              LISTENING    1756    PolicyAgent        [svchost.exe]
  TCP    [::]:49157         [::]:0              LISTENING    492     [lsass.exe]
```

The net use command shows the mapped drives in use by a system. Unknown or unfamiliar mapped drives should be investigated immediately for unauthorized activity and potential IOC. If there are other IOCs the machines that contain the mapped drives should be investigated. The net share command shows drives shared remotely along with all mapped drives in use. Unknown or unfamiliar shared and mapped drives should be investigated immediately for unauthorized activity. We did not have any drives mapped within our network, so neither of these commands yielded any output relevant for investigating a compromise.

The next command in the survey examines connectivity with respect to the NetBIOS protocol. The nbtstat –nrs command displays current NetBIOS cache information pertaining to remote machines from recent sessions. The type code identifies the purpose of the remote operating system within a network. Unusual recent sessions could lead to an IOC. NetBIOS type <20> is a file server service most likely associated with a recently established connection over port 445 [47]. Nbtstat is also used by attackers to enumerate vulnerable targets within a network. In the event that an IOC is found on a surveyed system, the output of nbtstat could reveal other compromised systems within the network [46, p. 20]. However, our survey of the infected machine produced no IOCs.

The next group of commands in the survey all focused on the services running on the computer. The net start command outputs a list of running services. Unknown, unfamiliar or recently launched services should be investigated as a potential IOC. The sc query command is a powerful tool similar to net start. It also allows for the ability to modify and display service configuration information. Particular information of interest is a service’s running state; “NOT_STOPPABLE” could be a method of persistence used by malware. Unknown or unfamiliar services should be investigated as a potential IOC. The psservice command outputs information similar to sc query; however, the added “DISPLAY_NAME” field provides the description and purpose of the service. Services with no write up, incorrect English, or inapplicable information should be investigated as a potential IOC. As Poison Ivy uses process injection and ADS to hide itself rather than manipulating running services, no IOCs were found in the outputs of these commands.

The next command run in the toolset is very useful when investigating a suspect process or service identified by tasklist, pslist, net start, or sc query. Autorunsc displays a list of currently configured auto-start applications and services. The output from autorunsc is beneficial because methods of persistence can be identified; if the autorunsc output of a system differs from its baseline and is not from an approved software installation or upgrade, this is a likely symptom of malware infection [46, pp. 44–45]. In our environment, this tool revealed no IOCs.

2. Registry Keys

The next group of tools examines settings within the Windows registry. The registry is a database of all of the settings and options within the Windows operating systems. It contains a large amount of data within a large number of keys and sub keys. Because of the registry’s complexity it is often used by attackers as a way to hide their attacks and maintain persistence. Microsoft’s Technet website is an invaluable resource when investigating suspicious entries within the registry. These tools focus on some of the most common registry keys that are used by attackers. We queried the registry for each key listed in the following paragraphs. None of them produced any IOCs, however, we provided the types of information that could present an IOC in each key.

The “FileRenameOperations” registry key can expose possible obfuscation techniques. This registry key keeps track of all file rename operations on a system. Any entries in this registry key should be investigated as they could be a possible IOC. In this scenario Poison Ivy used default install methods with no file rename operations for obfuscation. Had the RAT performed a file rename operation the registry key above would be populated with the original and new file names along with the time of the change. There is a sub key called “PendingFileRenameOperations.” This key lists any files to be renamed at the next system restart. This could be an IOC as well. If the malware cannot rename a file due to it being used by the operating system or a program, it will remain in this key until the next reboot. Since no files on the system had been renamed there was no output for this command.

The “Environment” registry key contains the path folder locations and path extension specifics. This is the default location from which certain executables are run by a user. Irregular system paths and path extensions can be indicative of obfuscation and persistence techniques. An example would be when unknown executable types are intentionally specified to execute within a shell from a set path. There are a large number of sub-keys in this listing, making determination of a compromise difficult. Questionable sub-keys and settings can be compared to Microsoft’s recognized environmental variable settings on the Technet site.

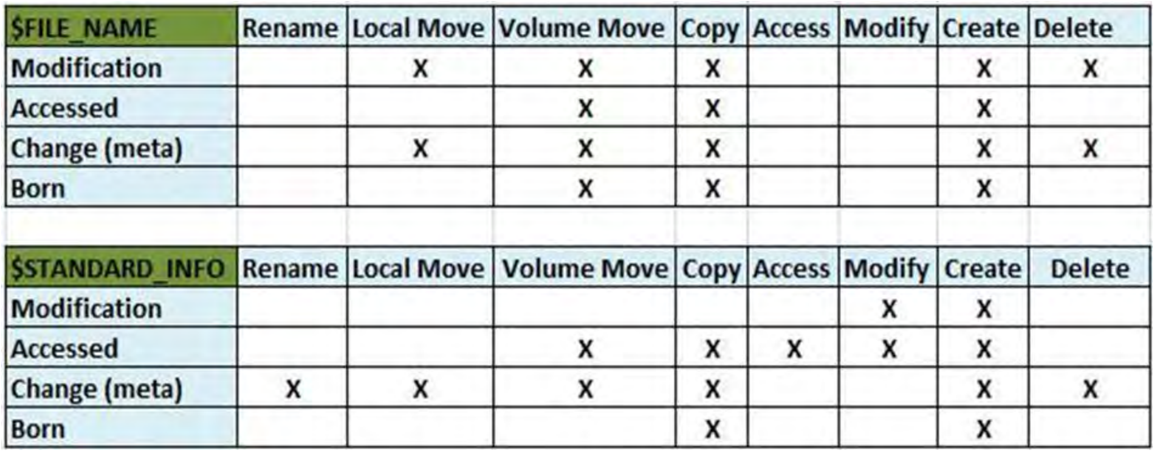
The “Hivelist” registry key contains a listing of the top level registry keys that are not recreated each time the system starts. These include the HKLM security, software and system hives. Any keys outside of these could be attempts at persistence by malware and should be investigated as a possible IOC. The output from the compromised machine contained no IOC and has been validated as normal on Microsoft’s development network [48].

The “CrashControl” registry key contains the path and file name for crashdump information. It also contains the settings for what actions Windows takes when a system crashes to include what information if any is written into the “crashdump” file. The “crashdump” file could also be used as a data exfiltration tactic in conjunction with a man-in-the-middle since the file contents are transmitted to Microsoft once connected to the Internet. Any sub keys that specify data locations outside of kernel and memory dumps should be treated as a possible IOC.

The “Winlogon” registry key’s location differs among versions of Windows. This registry key, along with its sub-keys governs all facets of access to a Windows system. The names of the registry sub-keys should match with the DLL files in the System32/ config directory. If they do not match DLL files in that directory it is possibly an IOC. This registry key can also be used for a few methods of malware persistence, such as the shell and userinit sub keys being replaced with malicious executables versus the genuine explore.exe or userinit.exe.

3. Metadata

Analysis of the Windows NTFS metadata can assist in discovering IOCs. It can also assist in building a timeline of events as the occurred during the compromise. This in turn can be used to aid in forensic analysis within this system and possibly on additional systems within the network. Figure 7 illustrates the file system operations that will update the timestamps stored as NTFS metadata. There are four date and time stamps in the $STANDARD_INFORMATION attribute [49, p. 317]. Our focus for this project is on three of them. The file creation (C), modification (W), and access (A) times can indicate when a file has been introduced to the system or when a file was last altered.



$FILE_NAME	Rename	Local Move	Volume Move	Copy	Access	Modify	Create	Delete
Modification		X	X	X			X	X
Accessed			X	X			X	
Change (meta)		X	X	X			X	X
Born			X	X			X	

$STANDARD_INFO	Rename	Local Move	Volume Move	Copy	Access	Modify	Create	Delete
Modification						X	X	
Accessed			X	X	X	X	X	
Change (meta)	X	X	X	X			X	X
Born				X			X	

Figure 7. File times and operations that update them, from [52]

The c:\windows\system32 directory contains many of the executables and DLLs associated with the Windows operating system. The files and folders contained within it should not typically be written to. Any entries with unusual timestamps should be examined as potential IOC. Table 16 presents the outputs of these commands from the baseline and infected scan. The highlighted entries represent an IOC that has not been obfuscated. The server.exe file is a recent and unknown addition to the system32 directory.

Table 16. File times for c:\windows\sytem32

Baseline Output	Infected Output
Directory of C:\windows\system32 06/10/2009 02:13 PM 3,698,584 ieapfltr.dat 08/18/2014 04:16 AM <DIR> drivers 08/19/2014 12:10 AM <DIR> config 08/22/2014 10:07 AM 9,792 7B296FB0-376B-497e-B012-9C450E1B7327-5P-1.C7483456-A289-439d-8115-601632D005A0 08/22/2014 10:07 AM 9,792 7B296FB0-376B-497e-B012-9C450E1B7327-5P-0.C7483456-A289-439d-8115-601632D005A0 2721 File(s) 971,004,466 bytes 89 Dir(s) 13,544,857,600 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289 -- Directory of C:\windows\system32 06/10/2009 02:13 PM 445,952 ieapfltr.dll 08/14/2014 04:48 AM 3,412 ALPHAWS01-2014-08-2014-Time- 08/15/2014 12:09 PM <DIR> . 08/15/2014 12:09 PM <DIR> .. 08/18/2014 04:16 AM <DIR> drivers 08/19/2014 12:10 AM <DIR> config 2721 File(s) 971,004,466 bytes 89 Dir(s) 13,544,857,600 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289 -- Directory of C:\windows\system32 06/10/2009 02:13 PM 445,952 ieapfltr.dll 08/14/2014 04:48 AM 3,412 ALPHAWS01-2014-08-2014-Time- 2721 File(s) 971,004,466 bytes 89 Dir(s) 13,544,857,600 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289	Directory of C:\windows\system32 06/10/2009 02:13 PM 3,698,584 ieapfltr.dat 08/14/2014 04:48 AM 3,412 ALPHAWS01-2014-08-2014-Time- 08/19/2014 12:10 AM <DIR> config 08/22/2014 12:40 PM 103,496 perfc009.dat 08/22/2014 12:40 PM 615,122 perfh009.dat 08/22/2014 12:40 PM 713,888 PerfStringBackup.INI 08/22/2014 01:27 PM <DIR> .. 08/22/2014 01:27 PM <DIR> . 08/22/2014 01:27 PM 9,216 server.exe 08/22/2014 01:33 PM <DIR> drivers 08/23/2014 12:38 PM 9,792 7B296FB0-376B-497e-B012-9C450E1B7327-5P-0.C7483456-A289-439d-8115-601632D005A0 08/23/2014 12:38 PM 9,792 7B296FB0-376B-497e-B012-9C450E1B7327-5P-1.C7483456-A289-439d-8115-601632D005A0 2722 File(s) 971,013,682 bytes 89 Dir(s) 13,539,770,368 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289 --- Directory of C:\windows\system32 06/10/2009 02:13 PM 445,952 ieapfltr.dll <...content removed for write up...> 08/14/2014 04:48 AM 3,412 ALPHAWS01-2014-08-2014-Time- 08/19/2014 12:10 AM <DIR> config 08/22/2014 01:27 PM 9,216 server.exe 08/22/2014 01:27 PM <DIR> .. 08/22/2014 01:27 PM <DIR> . 08/22/2014 01:33 PM <DIR> drivers 2722 File(s) 971,013,682 bytes 89 Dir(s) 13,539,770,368 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289 --- Directory of C:\windows\system32 06/10/2009 02:13 PM 445,952 ieapfltr.dll <...content removed for write up...> 08/14/2014 04:48 AM 3,412 ALPHAWS01-2014-08-2014-Time- 08/22/2014 01:27 PM 9,216 server.exe 2722 File(s) 971,013,682 bytes 89 Dir(s) 13,539,770,368 bytes free Volume in drive C has no label. Volume Serial Number is B64D-6289 Volume Serial Number is DE5C-61AC

B. RESPONSE AND RECOVERY

Upon completion of the analysis of the toolkit's output we determined that we had enough indications that the ALPHAWS01 had been compromised. After this determination was made, we took two initial actions. The first was on the production domain ALPHAWS01 computer. We moved it from the ALPHA VLAN to the ALPHA MIRROR VLAN. Then in the vSphere software we renamed the virtual machine to COMPROMISED_ALPHAWS01 to distinguish it as an infected machine. Next, we powered on the ALPHAWS01 machine in the mirror domain and moved it from the ALPHA MIRROR VLAN to the ALPHA VLAN.

The compromised machine was now homed in the mirror domain and no longer had access to any of the machines or resources in the production domain. Conversely, the mirror domain machine was now in the production domain. The mirrored domain machine is a copy of the machine that was created directly after installation and configuration. This means that all of the domain membership and access information carries over and the addition was seamless, without risk of continued compromise. At this point the system administrator still had access to the compromised machine in isolation. This serves two purposes.

First it allows the administrator to further investigate the machine. Additional information could indicate further compromise of the machine. The machine can be kept as it is indefinitely if a more detailed investigation is required. In a physical network this would mean a loss of the machine until an investigation was closed. However, with this implementation, there is no impact to the operation of the network. Also, this allows the system administrator to retrieve any critical data stored on the compromised machine in a controlled manner. Further investigation can indicate when the initial compromise occurred. This date can be used as a reference for restoring data from backups. Finally, when a compromise on a network has been determined, the possibility exists for other elements of the network to be compromised as well. Confirmation of malware on one machine requires a scan using the toolset across the entire enterprise. Any compromised machines would then be moved to the mirrored VLAN, with their counterpart moved to the production VLAN. With the compromised machine discovered and the initial triage performed, steps must then be taken remediate the conditions that allowed for the compromise; or the other machines on the network will remain vulnerable.

VII. CONCLUSION

(1) Summary

In this capstone project we created a toolset for the identification and classification of computer security compromises. In order to facilitate its use by Navy network administrators, it contains only programs native to Windows operating systems and the Sysinternals suite, which are already present on Navy networks. We then demonstrated the capability of this toolset to detect malware inserted into a computer in the network. With the compromise determined, we then examined how the use of mirrored VLANs and domains can facilitate an accelerated recovery from an incident. The incorporation of a native toolset and mirrored VLANS could provide Navy network administrators with an improved strategy for identifying, classifying, and recovering from compromises.

(2) Future work

There are further areas of development that could increase the usefulness of this project. The scenario presented here provides an excellent opportunity for adoption into coursework in the areas of forensics and incident response. The nature of this scenario would lend itself well to the examination of the toolset outputs for academic purposes in a lab exercise.

If we were to continue with this project the next phase would involve the Linux network we developed. An initial toolset for detecting compromises in a Linux network was developed but we were unable to test its efficacy due to time constraints. The list of these tools is available in Appendix E. Malware would need to be inserted into the Linux network, and the output from these tools would need to be examined to determine if they would provide reliable indicators of the compromise.

Finally, this project could be expanded through the use of other forms of malware and the investigation of other potential tools that could be included in the toolset. The use of other types of malware could be used to confirm the capability of these tools to provide an accurate status of a computer’s security. As more malware is analyzed using this toolset, a database of discovered IOCs could be created as a reference for use with the toolset. At the same time, the toolset we have created is not a static object. More research could prove that there are other tools available that would provide increased capabilities in performing remote triage on a network.

APPENDIX A. INSTALLING AND CONFIGURING DD-WRT ROUTING AND SWITCHING

A. INSTALLING THE IMAGE

1. On a fully functional Windows 7 machine within the vSphere environment browse to ftp://ftp.dd-wrt.com/others/eko/BrainSlayer-V24-preSP2/2014/06-23-2014-r24461/x86/

2. From here download the file dd-wrt_public_vga.image

3. Next, point the browser to http://m0n0.ch/wall/physdiskwrite.php

4. Download physdiskwrite 0.5.3

5. Shutdown the Windows 7 machine.

6. In the vSphere client software select File→New→Virtual Machine

7. Select the custom radio button

8. Name the machine

9. Select the appropriate storage location

10. Leave the default machine version

11. Select the Linux radio button, from the drop down menu select Other Linux 2.6 32-bit

12. Leave defaults for processors and RAM, these can be lowered if necessary due to limited resources

13. Select 2 NICs, both configured as E1000, the first NIC will be the WAN, the second will be the LAN

14. Leave the SCSI settings at the default

15. On the next page select do not create disk at this time

16. Select the check box to edit machine settings before completion and click continue

17. Click the Add button on the Virtual Machine properties window

18. Select hard disk, click next

19. Click next

20. Set disk size to 1 GB and click next

21. Select the radio button for IDE 0 : 0, click next

22. Click finish twice

23. Right click on the Windows 7 virtual machine that you downloaded the image and software to and select edit settings

24. Click add

25. Select hard disk and click next

26. Select use and existing virtual disk and click next

27. Browse the data store for the hard disk for the virtual machine just created, click ok, the click next through the rest of the prompts

28. Power on the Windows 7 virtual machine

29. Run physdiskwrite as an administrator

30. Right click on the new IDE hard drive, select “image laden” and “offlen” (the application is written in German)

31. Browse to the DD-WRT image. Select all file types at the bottom.

32. Select the image file and click open

33. Click ok and the click yes on the dialog boxes.

34. Click ok. The image is now on the virtual IDE drive.

35. Power off the Windows 7 machine.

36. In the vSphere software, right click on the Windows 7 virtual machine and select edit settings

37. Select the IDE virtual hard disk and click remove

38. Click ok

39. Power on the DD-WRT machine and begin the process of configuring the routing and switching.

B. CONFIGURING ROUTING AND SWITCHING

The default IP address for the LAN side NIC on a DD-WRT router is 192.168.1.1. All of the configurations are done via the web interface. To access this, log onto the Windows 7 virtual machine. Ensure the machine is configured for DHCP and that it is in the same vSphere VLAN as the second NIC on the DD-WRT machine.

1. Log into the Windows 7 virtual machine
2. Browse to 192.168.1.1
3. Set the user name and password
4. Set the WAN IP for manual configuration and fill in the router name and WAN IP address
5. Set the LAN IP, gateway and DNS
6. Save the settings
7. Click on the Networking link.
8. Add two VLANs in the VLAN subsection
9. Save settings

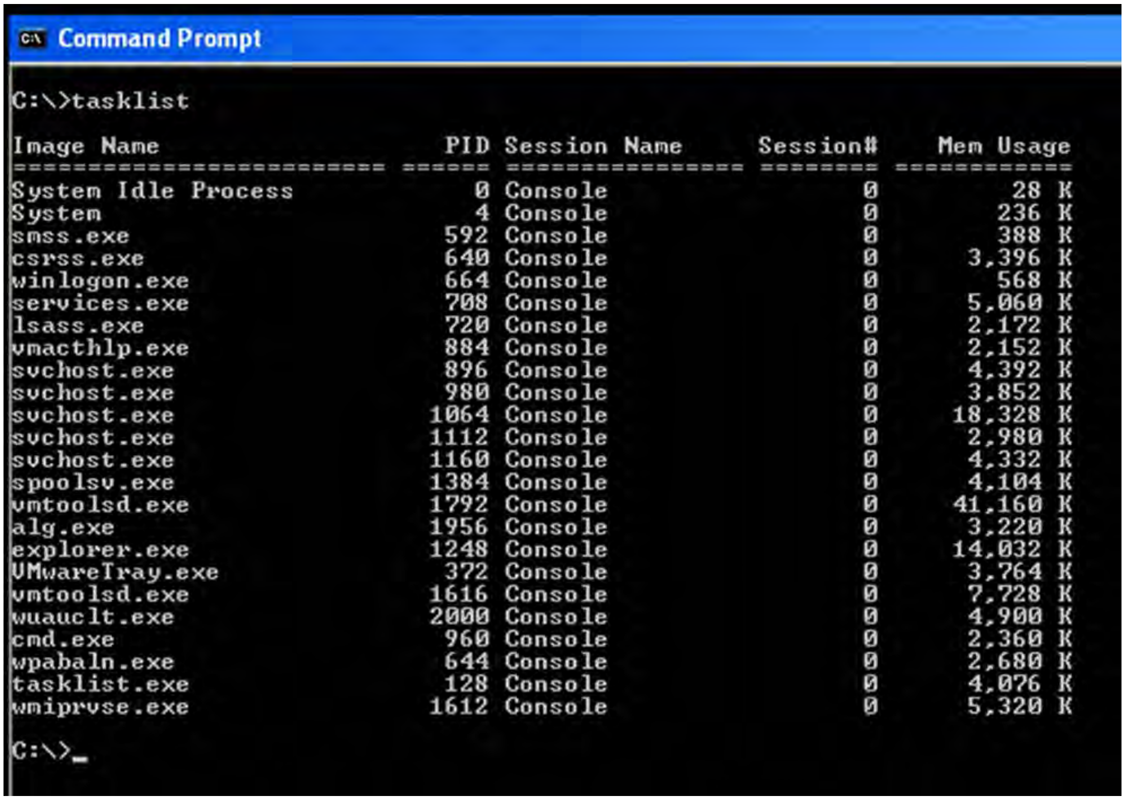
THIS PAGE LEFT INTENTIONALLY BLANK

APPENDIX B. WINDOWS PROCESSES

In this appendix we present some of the more common processes in the various Windows operating system environments. Many of the tools we used examined processes for potential IOCs. Recognizing the common Windows processes and their basic function aided in the analysis phase.

A. XP

Figure 8 below is a screen capture of the tasklist output from one of the Windows XP machines in our network. It shows many of the common processes for this OS. Table 16 then provides the functions of these processes.



```
C:\>tasklist

Image Name                   PID Session Name     Session#    Mem Usage
========================= ====== ================ ======== ============
System Idle Process            0 Console                 0         28 K
System                         4 Console                 0        236 K
smss.exe                     592 Console                 0        388 K
csrss.exe                    640 Console                 0      3,396 K
winlogon.exe                 664 Console                 0        568 K
services.exe                 708 Console                 0      5,060 K
lsass.exe                    720 Console                 0      2,172 K
vmacthlp.exe                 884 Console                 0      2,152 K
svchost.exe                  896 Console                 0      4,392 K
svchost.exe                  980 Console                 0      3,852 K
svchost.exe                 1064 Console                 0     18,328 K
svchost.exe                 1112 Console                 0      2,980 K
svchost.exe                 1160 Console                 0      4,332 K
spoolsv.exe                 1384 Console                 0      4,104 K
vmtoolsd.exe                1792 Console                 0     41,160 K
alg.exe                     1956 Console                 0      3,220 K
explorer.exe                1248 Console                 0     14,032 K
VMwareTray.exe               372 Console                 0      3,764 K
vmtoolsd.exe                1616 Console                 0      7,728 K
wuauclt.exe                 2000 Console                 0      4,900 K
cmd.exe                      960 Console                 0      2,360 K
wpabaln.exe                  644 Console                 0      2,680 K
tasklist.exe                 128 Console                 0      4,076 K
wmiprvse.exe                1612 Console                 0      5,320 K

C:\>_
```

Figure 8. Typical Windows XP process list

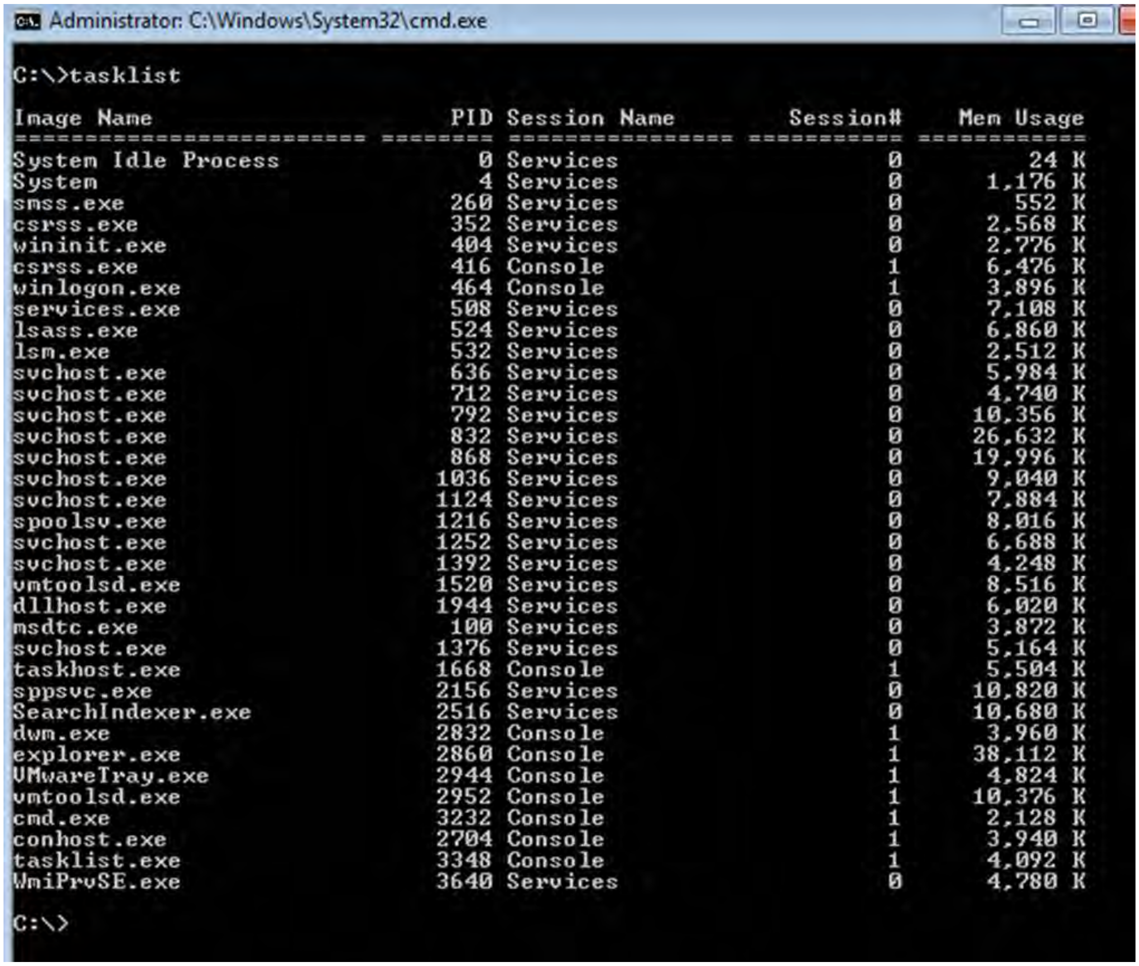
Table 16. Windows XP common processes and uses

alg.exe	The alg.exe (application layer gateway) allows applications (such as IM clients, RTSP, BitTorrent, SIP, and FTP) from a client computer to dynamically utilize passive TCP/ UDP ports in communicating with known ports on a server. This allows software to access applications that reside on another computer even if there is a firewall. The alg.exe file's absence would cause the security protocols to block communication ports, or for network administrators to consciously open numerous ports on the firewall that would create immense network vulnerabilities and potential threats.
cmd.exe	Enables execute of a batch file. An executable that provides the command prompt (MS-DOS shell interpreter) for Windows NT family.
csrss.exe	Needed to boot to Windows. Used to maintain the Win32 system environment console and other essential functions.
explorer.exe	The explorer.exe (located in the C:\WINDOWS folder), manages the Windows Graphical Shell including the Start Menu, Taskbar, Desktop, and File Manager. Without it running, the graphical interface for Windows will disappear. (The iertutil.dll is installed by Internet Explorer 7.)
lsass.exe	(LSA Security Service) - Needed to boot to Windows. The Local Security Authority server process. lsass .exe is a Lite belonging to Event Agent Setup from Event Agent
System Idle Process	System Idle Process is not a process, more a counter which is displayed in WinTasks used for measuring how much idle time the CPU is having at any particular time. This counter will display how much CPU Resources, as a percentage are 'idle' and available for use.
System	Non-system processes like [system process] originate from software installed on the system.
smss.exe	(Windows NT Session Manager) - Needed to boot to Windows. Used to establish the Windows XP environment during boot up. smss.exe is a process which is a part of the Microsoft Windows Operating System. It is called the Session Manager Subsystem and is responsible for handling sessions on the system.
services.exe	Services and Controller app) - Needed to boot to Windows. Main Service file for Plug and Play. Services.exe is a part of the Microsoft Windows Operating System and manages the operation of starting and stopping services. This process also deals with the automatic starting of services during the

	computer’s boot-up and the stopping of services during shut-down.
svchost.exe	(Generic Host Process for Win32 Services) - Needed to boot to Windows. The file svchost.exe is the Generic Host Process for Win32 Services used for administering 16-bit-based dynamically linked library files (DLL files) including other supplementary support applications.
spoolsv.exe	The spoolsv.exe file is described as the Spooler SubSystem App or Windows Print Spooler Service and is the main component of the printing interfaces. The spoolsv.exe file is initialized when the computer starts, and it runs in the background until the computer is turned off.
tasklist.exe	tasklist.exe displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.
vmacthlp.exe	VMware Physical Disk Help Service
vmtoolsd.exe	The VMware Tools service (vmtoolsd.exe on Windows guests or vmtoolsd on Linux and Solaris guests). This service synchronizes the time in the guest operating system with the time in the host operating system. On Windows guests, it also controls grabbing and releasing the mouse cursor.
VMwareTray.exe	VMwareTray.exe provides quick access to the most important program functions.
WmiPrvSE.exe	The wmiprvse.exe file is otherwise known as Windows Management Instrumentation. It is a Microsoft Windows-based component that provides control and information about management in an enterprise environment. Developers use the wmiprvse.exe file in order to develop applications used for monitoring purposes.
winlogon.exe	Windows Logon Application (Windows logon manager). H**a**ndles the login and logout procedures.
Wpabaln.exe	wpabaln.exe forms a part of the Microsoft Windows Operating System and is responsible for licensing issues on the computer.
Wuauclt.exe	Wuauclt.exe is the AutoUpdate Client of Windows Update and is used to check for available updates (for the various versions of the MS Windows platform) from Microsoft Update.

B. WINDOWS 7

Figure 9 below is a screen capture of the tasklist output from one of the Windows 7 machines in our network. It shows many of the common processes for this OS. Table 17 then provides the functions of these processes.



```
C:\>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0      1,176 K
smss.exe                       260 Services                   0        552 K
csrss.exe                      352 Services                   0      2,568 K
wininit.exe                    404 Services                   0      2,776 K
csrss.exe                      416 Console                    1      6,476 K
winlogon.exe                   464 Console                    1      3,896 K
services.exe                   508 Services                   0      7,108 K
lsass.exe                      524 Services                   0      6,860 K
lsm.exe                        532 Services                   0      2,512 K
svchost.exe                    636 Services                   0      5,984 K
svchost.exe                    712 Services                   0      4,740 K
svchost.exe                    792 Services                   0     10,356 K
svchost.exe                    832 Services                   0     26,632 K
svchost.exe                    868 Services                   0     19,996 K
svchost.exe                   1036 Services                   0      9,040 K
svchost.exe                   1124 Services                   0      7,884 K
spoolsv.exe                   1216 Services                   0      8,016 K
svchost.exe                   1252 Services                   0      6,688 K
svchost.exe                   1392 Services                   0      4,248 K
vmtoolsd.exe                  1520 Services                   0      8,516 K
dllhost.exe                   1944 Services                   0      6,020 K
msdtc.exe                      100 Services                   0      3,872 K
svchost.exe                   1376 Services                   0      5,164 K
taskhost.exe                  1668 Console                    1      5,504 K
sppsvc.exe                    2156 Services                   0     10,820 K
SearchIndexer.exe             2516 Services                   0     10,680 K
dwm.exe                       2832 Console                    1      3,960 K
explorer.exe                  2860 Console                    1     38,112 K
VMwareTray.exe                2944 Console                    1      4,824 K
vmtoolsd.exe                  2952 Console                    1     10,376 K
cmd.exe                       3232 Console                    1      2,128 K
conhost.exe                   2704 Console                    1      3,940 K
tasklist.exe                  3348 Console                    1      4,092 K
WmiPrvSE.exe                  3640 Services                   0      4,780 K

C:\>
```

Figure 9. Typical Windows 7 process list

Table 17. Common Windows 7 processes and uses

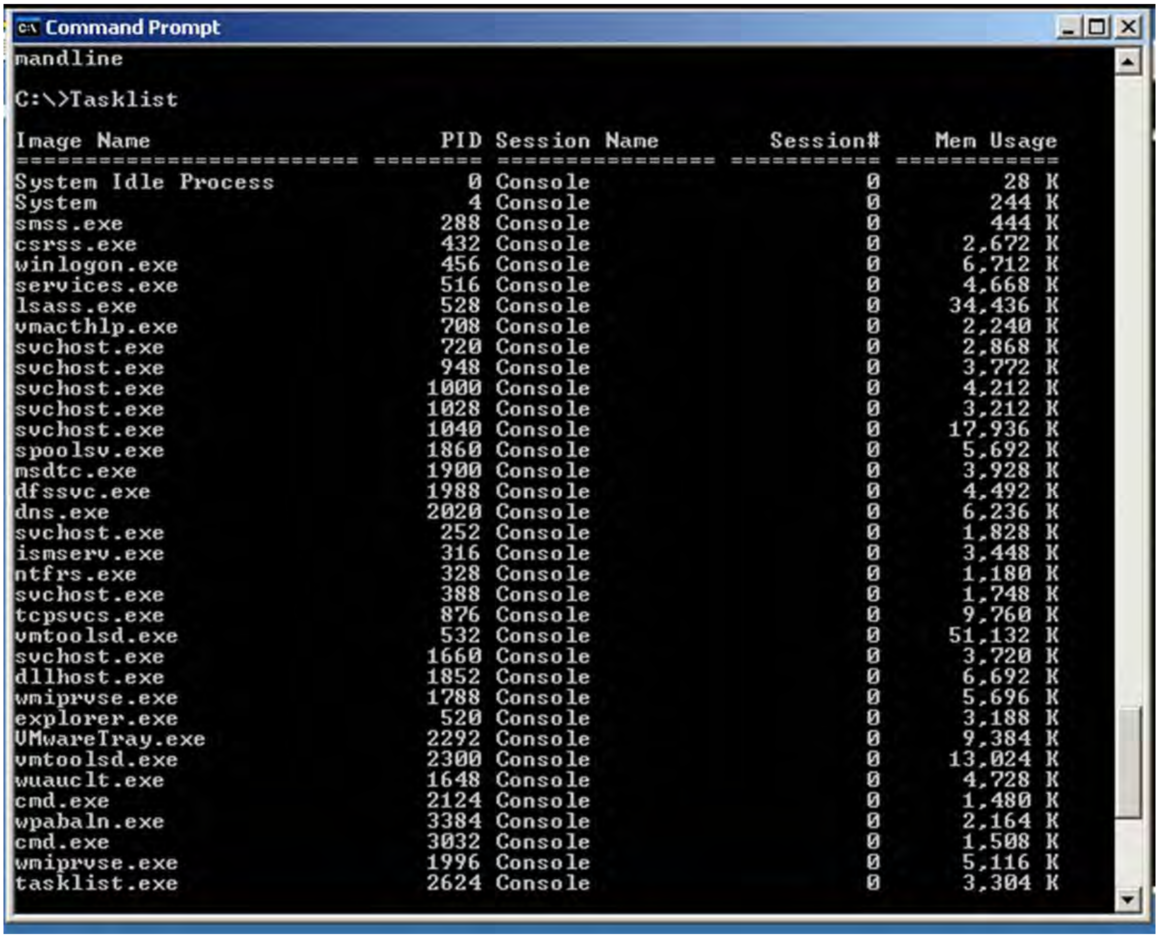
cmd.exe	Enables execute of a batch file. An executable that provides the command prompt (MS-DOS shell interpreter) for Windows NT family.
conhost.exe	conhost.exe is a Console Window Host, part of Microsoft Windows Operating System.
csrss.exe	Client-Server Runtime Server Subsystem- Needed to boot to Windows. Used to maintain the Win32 system environment console and other essential functions.
dwm.exe	dwm.exe is the Desktop Window Manager and is responsible for the graphical effects in Microsoft Windows Vista operating system such as 3D effects, live windows previews and windows transparencies.
dllhost.exe	dllhost.exe is a process belonging to Microsoft Windows Operating System. The dllhost.exe file manages DLL-based applications. This program is important for the stable and secure operation of the computer and should not be terminated.
explorer.exe	The explorer.exe (located in the C:\WINDOWS folder), manages the Windows Graphical Shell including the Start Menu, Taskbar, Desktop, and File Manager. Without it running, the graphical interface for Windows will disappear. (The iertutil.dll is installed by Internet Explorer 7.)
lsm.exe	Local Session Manager Service associated with the System Management task of the platform. In some platforms, it manages the connections related to the terminal server on the hosted machine.
lsass.exe	(LSA Security Service) - Needed to boot to Windows. The Local Security Authority server process. lsass .exe is a Lite belonging to Event Agent Setup from Event Agent
mstdc.exe	The Microsoft Distributed Transaction Coordinator is an application that is primarily used in allowing several other client applications to include more than one source of data in a single transaction.
SearchIndexer.exe	SearchIndexer.exe is the Windows service that handles indexing of files for Windows Search, which fuels the file search engine built into Windows that powers everything from the Start Menu search box to Windows Explorer, and even the Libraries feature.
System Idle Process	System Idle Process is not a process, more a counter which is displayed in WinTasks used for measuring how much idle time the CPU is having at any particular time. This counter will display how much CPU Resources, as a percentage are ‘idle’ and available for use.

System	Non-system processes like [system process] originate from software installed on the system.
smss.exe	(Windows NT Session Manager) - Needed to boot to Windows. Used to establish the Windows XP environment during bootup. smss.exe is a process which is a part of the Microsoft Windows Operating System. It is called the Session Manager Subsystem and is responsible for handling sessions on the system.
services.exe	(Services and Controller app) - Needed to boot to Windows. Main Service file for Plug and Play. services.exe is a part of the Microsoft Windows Operating System and manages the operation of starting and stopping services. This process also deals with the automatic starting of services during the computers boot-up and the stopping of services during shut-down.
svchost.exe	(Generic Host Process for Win32 Services) - Needed to boot to Windows. The file svchost.exe is the Generic Host Process for Win32 Services used for administering 16-bit-based dynamically linked library files (DLL files) including other supplementary support applications.
spoolsv.exe	The spoolsv.exe file is described as the Spooler SubSystem App or Windows Print Spooler Service and is the main component of the printing interfaces. The spoolsv.exe file is initialized when the computer starts, and it runs in the background until the computer is turned off.
sppsvc.exe	At startup, Windows automatically runs the sppsvc.exe file which creates the Software Protection Platform Service. This service is used by Windows 7 to (among other things) monitor its own protected system files to ensure they are not modified.
tasklist.exe	tasklist.exe displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.
taskhost.exe	The taskhost.exe is a Task Host which is a generic Host Process for Windows 7 32-bit Services. This generic process acts as a host for processes that run from DLLs. At startup of the system the TASKHOST checks the number of Services in the Registry and builds a list of DLL-based services which should be loaded and then loads them.
vmtoolsd.exe	The VMware Tools service (vmtoolsd.exe on Windows guests or vmtoolsd on Linux and Solaris guests). This service synchronizes the time in the guest operating system with the time in the host operating system. On Windows

	guests, it also controls grabbing and releasing the mouse cursor.
VMwareTray.exe	VMwareTray.exe provides quick access to the most important program functions.
WmiPrvSE.exe	The wmiprvse.exe file is otherwise known as Windows Management Instrumentation. It is a Microsoft Windows-based component that provides control and information about management in an enterprise environment. Developers use the wmiprvse.exe file in order to develop applications used for monitoring purposes.
winlogon.exe	Windows Logon Application (Windows logon manager). Handles the login and logout procedures.
wininit.exe	Wininit.exe was created to allow uninstallers to run and process commands stored in the file WinInit.ini. This allows programs to take action while the computer is still booting up. In Windows 7 and Vista, it primarily acts as a launcher for the majority of the background applications that are always running.

C. SERVER 2003

Figure 10 below is a screen capture of the tasklist output from one of the Windows Server 2003 machines in our network. It shows many of the common processes for this OS. Table 18 then provides the functions of these processes.



```
mandline

C:\>Tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Console                    0         28 K
System                           4 Console                    0        244 K
smss.exe                       288 Console                    0        444 K
csrss.exe                      432 Console                    0      2,672 K
winlogon.exe                   456 Console                    0      6,712 K
services.exe                   516 Console                    0      4,668 K
lsass.exe                      528 Console                    0     34,436 K
vmacthlp.exe                   708 Console                    0      2,240 K
svchost.exe                    720 Console                    0      2,868 K
svchost.exe                    948 Console                    0      3,772 K
svchost.exe                   1000 Console                    0      4,212 K
svchost.exe                   1028 Console                    0      3,212 K
svchost.exe                   1040 Console                    0     17,936 K
spoolsv.exe                   1860 Console                    0      5,692 K
msdtc.exe                     1900 Console                    0      3,928 K
dfssvc.exe                    1988 Console                    0      4,492 K
dns.exe                       2020 Console                    0      6,236 K
svchost.exe                    252 Console                    0      1,828 K
ismserv.exe                    316 Console                    0      3,448 K
ntfrs.exe                      328 Console                    0      1,180 K
svchost.exe                    388 Console                    0      1,748 K
tcpsvcs.exe                    876 Console                    0      9,760 K
vmtoolsd.exe                   532 Console                    0     51,132 K
svchost.exe                   1660 Console                    0      3,720 K
dllhost.exe                   1852 Console                    0      6,692 K
wmiprvse.exe                  1788 Console                    0      5,696 K
explorer.exe                   520 Console                    0      3,188 K
VMwareTray.exe                2292 Console                    0      9,384 K
vmtoolsd.exe                  2300 Console                    0     13,024 K
wuauclt.exe                   1648 Console                    0      4,728 K
cmd.exe                       2124 Console                    0      1,480 K
wpabaln.exe                   3384 Console                    0      2,164 K
cmd.exe                       3032 Console                    0      1,508 K
wmiprvse.exe                  1996 Console                    0      5,116 K
tasklist.exe                  2624 Console                    0      3,304 K
```

Figure 10. Typical Windows Server 2003 process list

Table 18. Common server 2003 processes and their purposes

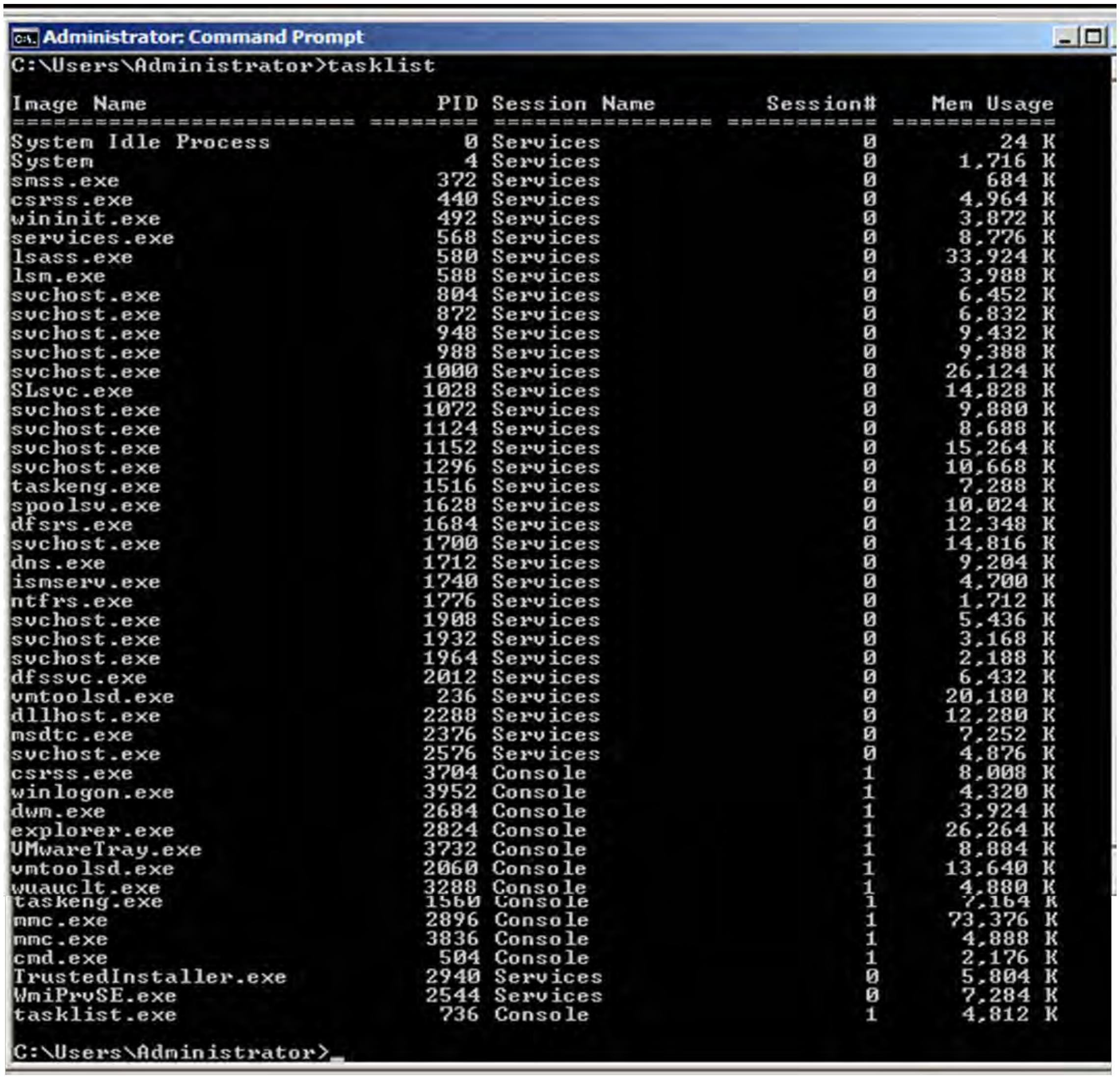
cmd.exe	Enables execute of a batch file. An executable that provides the command prompt (MS-DOS shell interpreter) for Windows NT family.
csrss.exe	Needed to boot to Windows. Used to maintain the Win32 system environment console and other essential functions.
dns.exe	dns.exe is the main process which handles the Microsoft Windows DNS server, if enabled. This program is important for the stable and secure running of the server and should not be terminated.
dfssvc.exe	dfssvc.exe is the Distributed File System service. Only found on Microsoft Windows Server suites, this process is crucial for the DFS service.

dllhost.exe	dllhost.exe is a process belonging to Microsoft Windows Operating System. The dllhost.exe file manages DLL based applications. This program is important for the stable and secure operation of the computer and should not be terminated.
explorer.exe	The explorer.exe (located in the C:\WINDOWS folder), manages the Windows Graphical Shell including the Start Menu, Taskbar, Desktop, and File Manager. Without it running, the graphical interface for Windows will disappear. (The iertutil.dll is installed by Internet Explorer 7.)
ismserv.exe	ismserv.exe allows messages to be sent via Microsoft Windows server sites. This is a non-essential process. Disabling or enabling it is a user preference.
lsass.exe	(LSA Security Service) - Needed to boot to Windows. The Local Security Authority server process. lsass .exe is a Lite belonging to Event Agent Setup from Event Agent
mstdc.exe	The Microsoft Distributed Transaction Coordinator is an application that is primarily used in allowing several other client applications to include more than one source of data in a single transaction.
ntfrs.exe	It is used to maintain file synchronization of file directory contents among multiple servers.
System Idle Process	System Idle Process is not a process, more a counter which is displayed in WinTasks used for measuring how much idle time the CPU is having at any particular time. This counter will display how much CPU Resources, as a percentage are ‘idle’ and available for use.
System	Non-system processes like [system process] originate from software installed on the system.
smss.exe	(Windows NT Session Manager) - Needed to boot to Windows. Used to establish the Windows XP environment during bootup. smss.exe is a process which is a part of the Microsoft Windows Operating System. It is called the Session Manager Subsystem and is responsible for handling sessions on the system.
services.exe	(Services and Controller app) - Needed to boot to Windows. Main Service file for Plug and Play. services.exe is a part of the Microsoft Windows Operating System and manages the operation of starting and stopping services.This process also deals with the automatic starting of services during the computer’s boot-up and the stopping of services during shut-down.
svchost.exe	(Generic Host Process for Win32 Services) - Needed to boot to Windows.

	The file svchost.exe is the Generic Host Process for Win32 Services used for administering 16-bit-based dynamically linked library files (DLL files) including other supplementary support applications.
spoolsv.exe	The spoolsv.exe file is described as the Spooler SubSystem App or Windows Print Spooler Service and is the main component of the printing interfaces. The spoolsv.exe file is initialized when the computer starts, and it runs in the background until the computer is turned off.
tasklist.exe	tasklist.exe displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.
tcpsvcs.exe	tcpsvcs.exe is a part of Microsoft Windows networking components. This essential system process is initiated when the computer uses special TCP/IP networking services such as DHCP, Simple TCP and print services.
vmtoolsd.exe	The VMware Tools service (vmtoolsd.exe on Windows guests or vmtoolsd on Linux and Solaris guests). This service synchronizes the time in the guest operating system with the time in the host operating system. On Windows guests, it also controls grabbing and releasing the mouse cursor.
vmacthlp.exe	VMware Physical Disk Help Service
VMwareTray.exe	VMwareTray.exe provides quick access to the most important program functions.
WmiPrvSE.exe	The wmiprvse.exe file is otherwise known as Windows Management Instrumentation. It is a Microsoft Windows-based component that provides control and information about management in an enterprise environment. Developers use the wmiprvse.exe file in order to develop applications used for monitoring purposes.
winlogon.exe	Windows Logon Application (Windows logon manager). Handles the login and logout procedures.
wininit.exe	Wininit.exe was created to allow uninstallers to run and process commands stored in the file WinInit.ini. This allows programs to take action while the computer is still booting up. In Windows 7 and Vista, it primarily acts as a launcher for the majority of the background applications that are always running.
Wpabaln.exe	wpabaln.exe forms a part of the Microsoft Windows Operating System and is responsible for licensing issues on the computer.
Wuauclt.exe	Wuauclt.exe is the AutoUpdate Client of Windows Update and is used to check for available updates (for the various versions of the MS Windows platform) from Microsoft Update.

D. SERVER 2008

Figure11 below is a screen capture of the tasklist output from one of the Windows Server 2008 machines in our network. It shows many of the common processes for this OS. Table 19 then provides the functions of these processes.



```
C:\Users\Administrator>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0      1,716 K
smss.exe                       372 Services                   0        684 K
csrss.exe                      440 Services                   0      4,964 K
wininit.exe                    492 Services                   0      3,872 K
services.exe                   568 Services                   0      8,776 K
lsass.exe                      580 Services                   0     33,924 K
lsm.exe                        588 Services                   0      3,988 K
svchost.exe                    804 Services                   0      6,452 K
svchost.exe                    872 Services                   0      6,832 K
svchost.exe                    948 Services                   0      9,432 K
svchost.exe                    988 Services                   0      9,388 K
svchost.exe                   1000 Services                   0     26,124 K
SLsvc.exe                     1028 Services                   0     14,828 K
svchost.exe                   1072 Services                   0      9,880 K
svchost.exe                   1124 Services                   0      8,688 K
svchost.exe                   1152 Services                   0     15,264 K
svchost.exe                   1296 Services                   0     10,668 K
taskeng.exe                   1516 Services                   0      7,288 K
spoolsv.exe                   1628 Services                   0     10,024 K
dfsrs.exe                     1684 Services                   0     12,348 K
svchost.exe                   1700 Services                   0     14,816 K
dns.exe                       1712 Services                   0      9,204 K
ismserv.exe                   1740 Services                   0      4,700 K
ntfrs.exe                     1776 Services                   0      1,712 K
svchost.exe                   1908 Services                   0      5,436 K
svchost.exe                   1932 Services                   0      3,168 K
svchost.exe                   1964 Services                   0      2,188 K
dfssvc.exe                    2012 Services                   0      6,432 K
vmtoolsd.exe                   236 Services                   0     20,180 K
dllhost.exe                   2288 Services                   0     12,280 K
msdtc.exe                     2376 Services                   0      7,252 K
svchost.exe                   2576 Services                   0      4,876 K
csrss.exe                     3704 Console                    1      8,008 K
winlogon.exe                  3952 Console                    1      4,320 K
dwm.exe                       2684 Console                    1      3,924 K
explorer.exe                  2824 Console                    1     26,264 K
VMwareTray.exe                3732 Console                    1      8,884 K
vmtoolsd.exe                  2060 Console                    1     13,640 K
wuauclt.exe                   3288 Console                    1      4,880 K
taskeng.exe                   1560 Console                    1      7,164 K
mmc.exe                       2896 Console                    1     73,376 K
mmc.exe                       3836 Console                    1      4,888 K
cmd.exe                        504 Console                    1      2,176 K
TrustedInstaller.exe          2940 Services                   0      5,804 K
WmiPrvSE.exe                  2544 Services                   0      7,284 K
tasklist.exe                   736 Console                    1      4,812 K

C:\Users\Administrator>_
```

Figure 11. Typical Windows Server 2008 process list

Table 19. Common server 2008 processes and their purposes

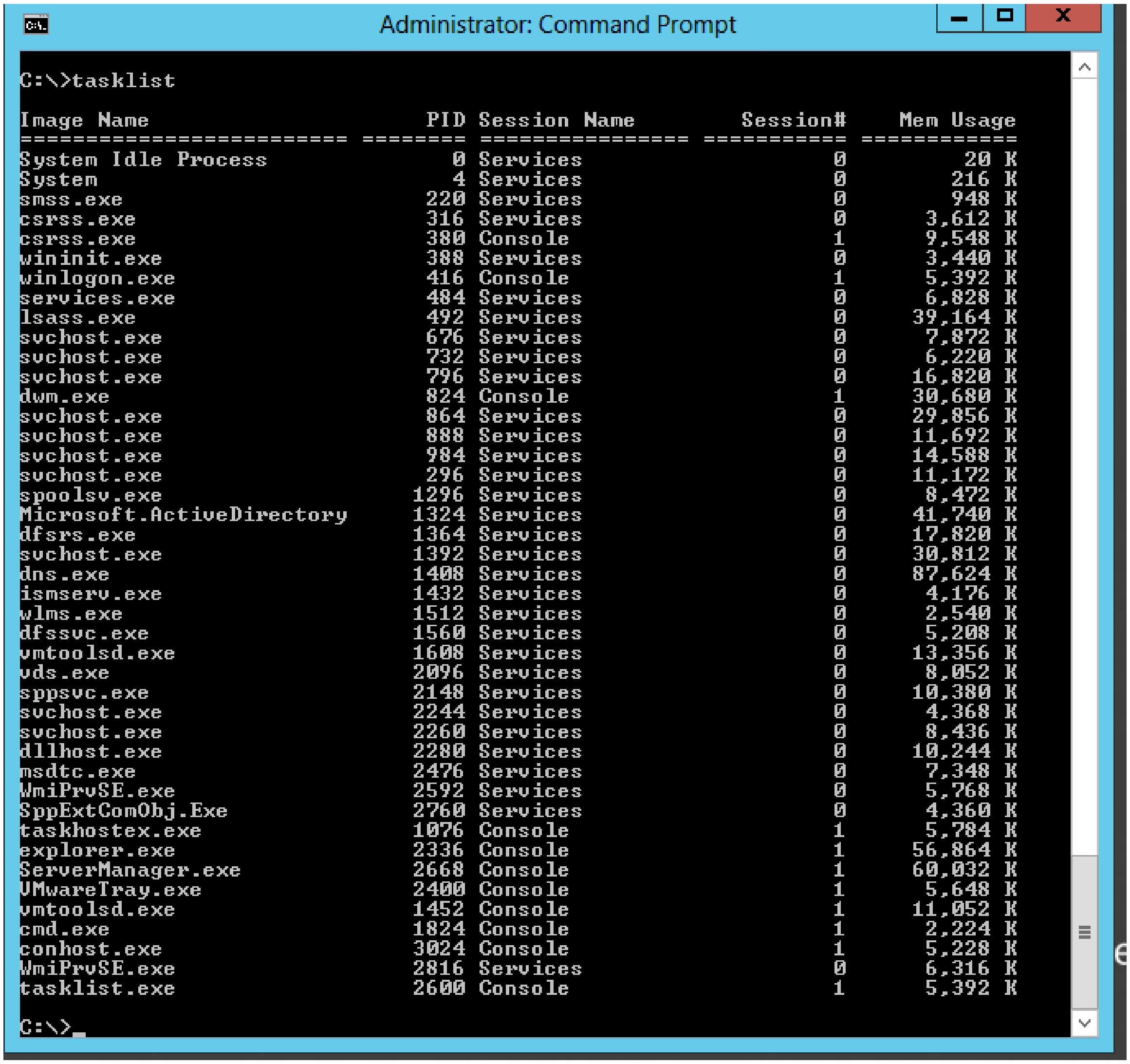
cmd.exe	Enables execute of a batch file. An executable that provides the command prompt (MS-DOS shell interpreter) for Windows NT family.
csrss.exe	Needed to boot to Windows. Used to maintain the Win32 system environment console and other essential functions.
dwm.exe	dwm.exe is the Desktop Window Manager and is responsible for the graphical effects in Microsoft Windows Vista operating system such as 3D effects, live windows previews and windows transparencies.
dfsrs.exe	Distributed File System Replication. A distributed file system is a system where network folders are accessed by users on a network but where the files inside those folders are not necessarily all on the same server - some files may be stored on one server, while other files on another server, etc.., but the entire folder appears as one folder to all users whichever server they connect to.
dns.exe	dns.exe is the main process which handles the Microsoft Windows DNS server, if enabled. This program is important for the stable and secure running of the server and should not be terminated.
dfssvc.exe	dfssvc.exe is the Distributed File System service. Only found on Microsoft Windows Server suites, this process is crucial for the DFS service.
dllhost.exe	dllhost.exe is a process belonging to Microsoft Windows Operating System. The dllhost.exe file manages DLL based applications. This program is important for the stable and secure operation of the computer and should not be terminated.
explorer.exe	The explorer.exe (located in the C:\WINDOWS folder), manages the Windows Graphical Shell including the Start Menu, Taskbar, Desktop, and File Manager. Without it running, the graphical interface for Windows will disappear. (The iertutil.dll is installed by Internet Explorer 7.)
ismserv.exe	ismserv.exe allows messages to be sent via Microsoft Windows server sites. This is a non-essential process. Disabling or enabling it is down to user preference.
lsm.exe	Local Session Manager Service associated with the System Management task of the platform. In some platform, it manages the connections related to the terminal server on the hosted machine.

lsass.exe	(LSA Security Service) - Needed to boot to Windows. The Local Security Authority server process. lsass .exe is a Lite belonging to Event Agent Setup from Event Agent
mstdc.exe	The Microsoft Distributed Transaction Coordinator is an application that is primarily used in allowing several other client applications to include more than one source of data in a single transaction.
Microsoft.ActiveDirectory	Active Directory is a special-purpose database.
mmc.exe	mmc.exe is the Microsoft Management Console application and is used to display various management plug-ins accessed from the Control Panel, such as the Device Manager.
ntfrs.exe	It is used to maintain file synchronization of file directory contents among multiple servers.
SLsvc.exe	slsvc.exe is a Software Licensing Service, used to protect digital products from copyright infringement.
System Idle Process	System Idle Process is not a process, more a counter which is displayed in WinTasks used for measuring how much idle time the CPU is having at any particular time. This counter will display how much CPU Resources, as a percentage are ‘idle’ and available for use.
System	Non-system processes like [system process] originate from software installed on the system.
smss.exe	(Windows NT Session Manager) - Needed to boot to Windows. Used to establish the Windows XP environment during bootup. smss.exe is a process which is a part of the Microsoft Windows Operating System. It is called the Session Manager Subsystem and is responsible for handling sessions on the system.
services.exe	(Services and Controller app) - Needed to boot to Windows. Main Service file for Plug and Play. services.exe is a part of the Microsoft Windows Operating System and manages the operation of starting and stopping services.This process also deals with the automatic starting of services during the computer’s boot-up and the stopping of services during shut-down.
svchost.exe	(Generic Host Process for Win32 Services) - Needed to boot to Windows. The file svchost.exe is the Generic Host Process for Win32 Services used for administering 16-bit-based dynamically linked library files (DLL files) including other supplementary support applications.
spoolsv.exe	The spoolsv.exe file is described as the Spooler SubSystem App or Windows Print Spooler Service and is

	the main component of the printing interfaces. The spoolsv.exe file is initialized when the computer starts, and it runs in the background until the computer is turned off.
tasklist.exe	tasklist.exe displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.
taskeng.exe	Task Scheduler Engine is responsible for running certain process at pre-defined times.
TrustedInstaller.exe	trustedinstaller.exe is a Windows Modules Installer and it also enables management of Windows updates.
vmtoolsd.exe	The VMware Tools service (vmtoolsd.exe on Windows guests or vmtoolsd on Linux and Solaris guests). This service synchronizes the time in the guest operating system with the time in the host operating system. On Windows guests, it also controls grabbing and releasing the mouse cursor.
VMwareTray.exe	VMwareTray.exe provides quick access to the most important program functions.
WmiPrvSE.exe	The wmiprvse.exe file is otherwise known as Windows Management Instrumentation. It is a Microsoft Windows-based component that provides control and information about management in an enterprise environment. Developers use the wmiprvse.exe file in order to develop applications used for monitoring purposes.
wininit.exe	Wininit.exe was created to allow uninstallers to run and process commands stored in the file WinInit.ini. This allows programs to take action while the computer is still booting up. In Windows 7 and Vista, it primarily acts as a launcher for the majority of the background applications that are always running.
Wuauclt.exe	Wuauclt.exe is the AutoUpdate Client of Windows Update and is used to check for available updates (for the various versions of the MS Windows platform) from Microsoft Update.

E. SERVER 2012

Figure 12 is a screen capture of the tasklist output from one of the Windows Server 2012 machines in our network. It shows many of the common processes for this OS. Table 20 then provides the functions of these processes.



```
C:\>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         20 K
System                           4 Services                   0        216 K
smss.exe                       220 Services                   0        948 K
csrss.exe                      316 Services                   0      3,612 K
csrss.exe                      380 Console                    1      9,548 K
wininit.exe                    388 Services                   0      3,440 K
winlogon.exe                   416 Console                    1      5,392 K
services.exe                   484 Services                   0      6,828 K
lsass.exe                      492 Services                   0     39,164 K
svchost.exe                    676 Services                   0      7,872 K
svchost.exe                    732 Services                   0      6,220 K
svchost.exe                    796 Services                   0     16,820 K
dwm.exe                        824 Console                    1     30,680 K
svchost.exe                    864 Services                   0     29,856 K
svchost.exe                    888 Services                   0     11,692 K
svchost.exe                    984 Services                   0     14,588 K
svchost.exe                    296 Services                   0     11,172 K
spoolsv.exe                   1296 Services                   0      8,472 K
Microsoft.ActiveDirectory     1324 Services                   0     41,740 K
dfsrs.exe                     1364 Services                   0     17,820 K
svchost.exe                   1392 Services                   0     30,812 K
dns.exe                       1408 Services                   0     87,624 K
ismserv.exe                   1432 Services                   0      4,176 K
wlms.exe                      1512 Services                   0      2,540 K
dfssvc.exe                    1560 Services                   0      5,208 K
vmtoolsd.exe                  1608 Services                   0     13,356 K
vds.exe                       2096 Services                   0      8,052 K
sppsvc.exe                    2148 Services                   0     10,380 K
svchost.exe                   2244 Services                   0      4,368 K
svchost.exe                   2260 Services                   0      8,436 K
dllhost.exe                   2280 Services                   0     10,244 K
msdtc.exe                     2476 Services                   0      7,348 K
WmiPrvSE.exe                  2592 Services                   0      5,768 K
SppExtComObj.Exe              2760 Services                   0      4,360 K
taskhostex.exe                1076 Console                    1      5,784 K
explorer.exe                  2336 Console                    1     56,864 K
ServerManager.exe             2668 Console                    1     60,032 K
VMwareTray.exe                2400 Console                    1      5,648 K
vmtoolsd.exe                  1452 Console                    1     11,052 K
cmd.exe                       1824 Console                    1      2,224 K
conhost.exe                   3024 Console                    1      5,228 K
WmiPrvSE.exe                  2816 Services                   0      6,316 K
tasklist.exe                  2600 Console                    1      5,392 K

C:\>_
```

Figure 12. Typical Windows Server 2012 process list

Table 20. Common server 2012 processes and their purposes

cmd.exe	Enables execute of a batch file. An executable that provides the command prompt (MS-DOS shell interpreter) for Windows NT family.
conhost.exe	conhost.exe is a Console Window Host, part of Microsoft Windows Operating System.
csrss.exe	Needed to boot to Windows. Used to maintain the Win32 system environment console and other essential functions.
dwm.exe	dwm.exe is the Desktop Window Manager and is responsible for the graphical effects in Microsoft Windows Vista operating system such as 3D effects, live windows previews and windows transparencies
dfsrs.exe	Distributed File System Replication. A distributed file system is a system where network folders are accessed by users on a network but where the files inside those folders are not necessarily all on the same server - some files may be stored on one server, while other files on another server, etc.., but the entire folder appears as one folder to all users whichever server they connect to.
dns.exe	dns.exe is the main process which handles the Microsoft Windows DNS server, if enabled. This program is important for the stable and secure running of the server and should not be terminated.
dfssvc.exe	dfssvc.exe is the Distributed File System service. Only found on Microsoft Windows Server suites, this process is crucial for the DFS service.
dllhost.exe	dllhost.exe is a process belonging to Microsoft Windows Operating System. The dllhost.exe file manages DLL based applications. This program is important for the stable and secure operation of the computer and should not be terminated.
explorer.exe	The explorer.exe (located in the C:\WINDOWS folder), manages the Windows Graphical Shell including the Start Menu, Taskbar, Desktop, and File Manager. Without it running, the graphical interface for Windows will disappear. (The iertutil.dll is installed by Internet Explorer 7.)
ismserv.exe	ismserv.exe allows messages to be sent via Microsoft Windows server sites. This is a non-essential process. Disabling or enabling it is down to user preference.
lsass.exe	(LSA Security Service) - Needed to boot to Windows. The Local Security Authority server process. lsass .exe is a Lite belonging to Event Agent Setup from Event Agent

mstdc.exe	The Microsoft Distributed Transaction Coordinator is an application that is primarily used in allowing several other client applications to include more than one source of data in a single transaction.
Microsoft.ActiveDirectory	Active Directory is a special-purpose database.
System Idle Process	System Idle Process is not a process, more a counter which is displayed in WinTasks used for measuring how much idle time the CPU is having at any particular time. This counter will display how much CPU Resources, as a percentage are ‘idle’ and available for use.
System	Non-system processes like [system process] originate from software installed on the system.
smss.exe	(Windows NT Session Manager) - Needed to boot to Windows. Used to establish the Windows XP environment during bootup. smss.exe is a process which is a part of the Microsoft Windows Operating System. It is called the Session Manager Subsystem and is responsible for handling sessions on the system.
services.exe	(Services and Controller app) - Needed to boot to Windows. Main Service file for Plug and Play. services.exe is a part of the Microsoft Windows Operating System and manages the operation of starting and stopping services.This process also deals with the automatic starting of services during the computers boot-up and the stopping of services during shut-down.
svchost.exe	(Generic Host Process for Win32 Services) - Needed to boot to Windows. The file svchost.exe is the Generic Host Process for Win32 Services used for administering 16-bit-based dynamically linked library files (DLL files) including other supplementary support applications.
spoolsv.exe	The spoolsv.exe file is described as the Spooler SubSystem App or Windows Print Spooler Service and is the main component of the printing interfaces. The spoolsv.exe file is initialized when the computer starts, and it runs in the background until the computer is turned off.
sppsvc.exe	At startup, Windows automatically runs the sppsvc.exe file which creates the Software Protection Platform Service. This service is used by Windows 7 to (among other things) monitor its own protected system files to ensure they are not modified.
SppExtComObj.Exe	sppextcomobj.exe is filename of the process running on Microsoft Windows operating system. File version

	information describes this process as KMS Connection Broker.
ServerManager.exe	ServerManager.exe enables users to perform automated installations or removals of roles, role services, and features. ServerManagerCmd.exe options enable users to view logs of its operations and to run queries to display lists of roles, role services, and features that are both installed and available for installation on a computer.
tasklist.exe	tasklist.exe displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.
taskhostex.exe	The taskhost.exe is a Task Host which is a generic Host Process for Windows 7 32-bit Services. This generic process acts as a host for processes that run from DLLs. At startup of the system the TASKHOST checks the number of Services in the Registry and builds a list of DLL-based services which should be loaded and then loads them.
vmtoolsd.exe	The VMware Tools service (vmtoolsd.exe on Windows guests or vmtoolsd on Linux and Solaris guests). This service synchronizes the time in the guest operating system with the time in the host operating system. On Windows guests, it also controls grabbing and releasing the mouse cursor.
vds.exe	vds.exe is a Virtual Disk Service and it Provides management services for disks, volumes, file systems, and storage arrays
VMwareTray.exe	VMwareTray.exe provides quick access to the most important program functions.
WmiPrvSE.exe	The wmiprvse.exe file is otherwise known as Windows Management Instrumentation. It is a Microsoft Windows-based component that provides control and information about management in an enterprise environment. Developers use the wmiprvse.exe file in order to develop applications used for monitoring purposes.
wlms.exe	Windows License Monitoring Service
winlogon.exe	Windows Logon Application (Windows logon manager). Handles the login and logout procedures.
wininit.exe	Wininit.exe was created to allow uninstallers to run and process commands stored in the file WinInit.ini. This allows programs to take action while the computer is still booting up. In Windows 7 and Vista, it primarily acts as a launcher for the majority of the background applications that are always running.

APPENDIX C. BATCH FILE CONTENTS

This appendix provides the contents of the two batch files we used to automate the deployment of the tools. The first tool is connect.bat. This batch file copies the second batch file toolscript.bat to the administrative share on every machine in the network. It then uses the psexec tool to remotely connect to each machine and execute the toolscript,bat file. The toolscript.bat file then runs each tool on the remote machines redirecting the output to file on the Triage machine.

A. CONNECT.BAT

```
cd c:\tools
copy c:\tools\toolscript.bat \\alphadc01\admin$
psexec \\alphadc01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphafs01\admin$
psexec \\alphafs01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaex01\admin$
psexec \\alphaex01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphawww1\admin$
psexec \\alphawww1 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws01\admin$
psexec \\alphaws01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws02\admin$
psexec \\alphaws02 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws03\admin$
psexec \\alphaws003 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws04\admin$
psexec \\alphaws04 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws05\admin$
psexec \\alphaws05 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\alphaws06\admin$
psexec \\alphaws06 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravodc01\admin$
psexec \\bravodc01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravoex01\admin$
psexec \\bravoex01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravofs01\admin$
psexec \\bravofs01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravowww1\admin$
psexec \\bravowww1 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows01\admin$
```

```
psexec \\bravows01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows02\admin$
psexec \\bravows02 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows03\admin$
psexec \\bravows03 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows04\admin$
psexec \\bravows04 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows05\admin$
psexec \\bravows05 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\bravows06\admin$
psexec \\bravows06 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliedc01\admin$
psexec \\charliedc01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliefs01\admin$
psexec \\charliefs01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charlieex01\admin$
psexec \\charlieex01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliewww1\admin$
psexec \\charliewww1 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews01\admin$
psexec \\charliews01 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews02\admin$
psexec \\charliews02 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews03\admin$
psexec \\charliews03 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews04\admin$
psexec \\charliews04 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews05\admin$
psexec \\charliews05 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
copy c:\tools\toolscript.bat \\charliews06\admin$
psexec \\charliews06 -u alpha\administrator -p P@$$w0rd c:\windows\toolscript.bat
```

B. TOOLSCRIPT.BAT

```
date /t >> \\triage\tools\results\%computername%.txt
time /t >> \\triage\tools\results\%computername%.txt
\\triage\tools\sigcheck -accepteula -e -u c:\windows\system32 >> \\triage\tools\results\%computername%.txt
\\triage\tools\psloggedon -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\logonsessions -p -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\net session >> \\triage\tools\results\%computername%.txt
\\triage\tools\pslist -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\tasklist >> \\triage\tools\results\%computername%.txt
\\triage\tools\listdlls -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\handle -accepteula >> \\triage\tools\results\%computername%.txt
```

```
\\triage\tools\ipconfig /all >> \\triage\tools\results\%computername%.txt
\\triage\tools\netstat -an >> \\triage\tools\results\%computername%.txt
\\triage\tools\net use >> \\triage\tools\results\%computername%.txt
\\triage\tools\net share >> \\triage\tools\results\%computername%.txt
\\triage\tools\nbtstat -nrs >> \\triage\tools\results\%computername%.txt
\\triage\tools\nbtstat -c >> \\triage\tools\results\%computername%.txt
\\triage\tools\net start >> \\triage\tools\results\%computername%.txt
\\triage\tools\sc query >> \\triage\tools\results\%computername%.txt
\\triage\tools\psservice -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\driverquery >> \\triage\tools\results\%computername%.txt
\\triage\tools\autorunsc -a -accepteula >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\Session
Manager” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\Session
Manager\Memory Management” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\Session
Manager\FileRenameOperations” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\Session
Manager\Environment” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\Hivelist” >>
\\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\System\CurrentControlSet\Control\CrashControl”
>> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\Software\Microsoft\Windows NT\CurrentVersion\
Component Based Servicing\Packages” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\Software\Microsoft\Windows NT\CurrentVersion\
Winlogon” >> \\triage\tools\results\%computername%.txt
\\triage\tools\reg.exe query “HKLM\Software\Microsoft\Windows NT\CurrentVersion\
AeDebug” >> \\triage\tools\results\%computername%.txt
dir /A /T:W /O:D C:\windows\Tasks >> \\triage\tools\results\%computername%.txt
dir /A /T:A /O:D C:\windows\Tasks >> \\triage\tools\results\%computername%.txt
dir /A /T:C /O:D C:\windows\Tasks >> \\triage\tools\results\%computername%.txt
dir /A /T:W /O:D C:\windows\ >> \\triage\tools\results\%computername%.txt
dir /A /T:A /O:D C:\windows\ >> \\triage\tools\results\%computername%.txt
dir /A /T:C /O:D C:\windows\ >> \\triage\tools\results\%computername%.txt
dir /A /T:W /O:D C:\windows\system32 >> \\triage\tools\results\%computername%.txt
dir /A /T:A /O:D C:\windows\system32 >> \\triage\tools\results\%computername%.txt
dir /A /T:C /O:D C:\windows\system32 >> \\triage\tools\results\%computername%.txt
dir /A /T:W /O:D C:\windows\system32\drivers >> \\triage\tools\
results\%computername%.txt
dir /A /T:A /O:D C:\windows\system32\drivers >> \\triage\tools\
results\%computername%.txt
dir /A /T:C /O:D C:\windows\system32\drivers >> \\triage\tools\
results\%computername%.txt
```

```
dir /A /T:C /O:D C:\windows\system32\config >> \\triage\tools\results\%computername%.txt
dir /A /T:A /O:D C:\windows\system32\config >> \\triage\tools\results\%computername%.txt
dir /A /T:W /O:D C:\windows\system32\config >> \\triage\tools\results\%computername%.txt
doskey /h >> \\triage\tools\results\%computername%.txt
```

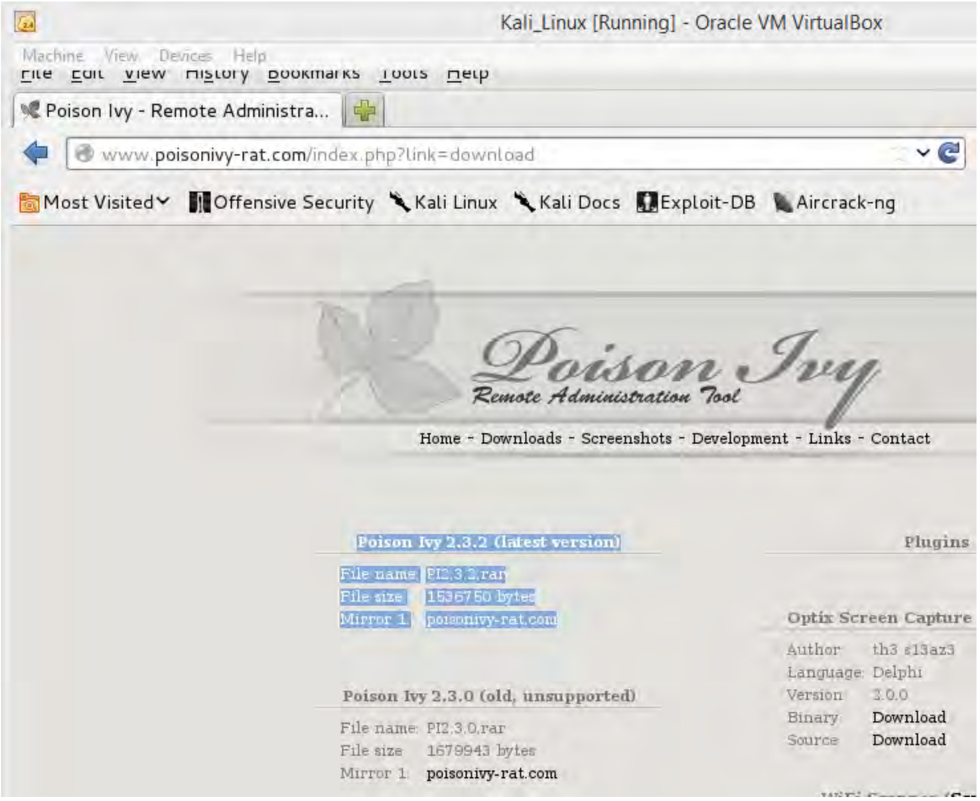
APPENDIX D. INSTALLATION AND CONFIGURATION OF POISON IVY

This appendix provides the steps we took to install and configure Poison Ivy for this capstone project.

A. INSTALLING POISON IVY

1. Point the browser to http://www.poisonivy-rat.com/index.php?link=download

2. Figure 13 shows the Poison Ivy website and where to download Poison Ivy 2.3.2


Kali_Linux [Running] - Oracle VM VirtualBox
Machine View Devices Help
File Edit View History Bookmarks Tools Help
Poison Ivy - Remote Administra...
www.poisonivy-rat.com/index.php?link=download
Most Visited
Offensive Security
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng
Poison Ivy
Remote Administration Tool
Home - Downloads - Screenshots - Development - Links - Contact
Plugins

Poison Ivy 2.3.2 (latest version)

File name	PI2.3.2.rar
File size	1536750 bytes
Mirror 1	poisonivy-rat.com

Poison Ivy 2.3.0 (old, unsupported)

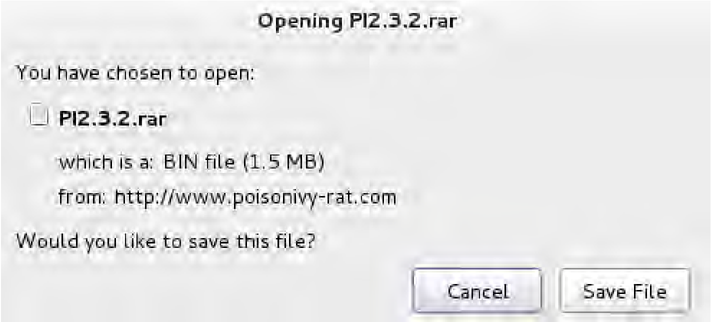
File name	PI2.3.0.rar
File size	1679943 bytes
Mirror 1	poisonivy-rat.com

Optix Screen Capture

Author	th3 s13az3
Language	Delphi
Version	3.0.0
Binary	Download
Source	Download

Figure 13. Poison Ivy website

3. Click next to mirror 1 to download, and click save file when the dialog box shown in Figure 14 appears.



Opening PI2.3.2.rar

You have chosen to open:

PI2.3.2.rar

which is a: BIN file (1.5 MB)

from: http://www.poisonivy-rat.com

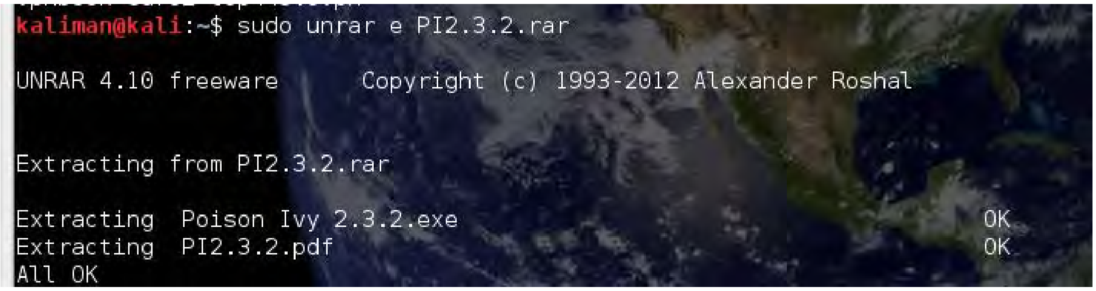
Would you like to save this file?

Figure 14. Pop-up dialog box

4. For Linux, to extract .rar files (Roshal ARchive)

5. Run sudo apt-get install unrar

6. Type unrar e PI2.3.2.rar. Figure 15 shows this command in progress.



```
kaliman@kali:~$ sudo unrar e PI2.3.2.rar

UNRAR 4.10 freeware      Copyright (c) 1993-2012 Alexander Roshal


Extracting from PI2.3.2.rar

Extracting  Poison Ivy 2.3.2.exe                                    OK
Extracting  PI2.3.2.pdf                                             OK
All OK
```

Figure 15. Unzipping the Poison Ivy package

B. BUILDING POISON IVY SERVER

1. DoubleClick on the Poison Ivy 2.3.2.exe and accept the Terms and Conditions EULA.

2. Select “File” -> “New Server”

3. Click “Create Profile” as shown in Figure 16 -> create User ID and password

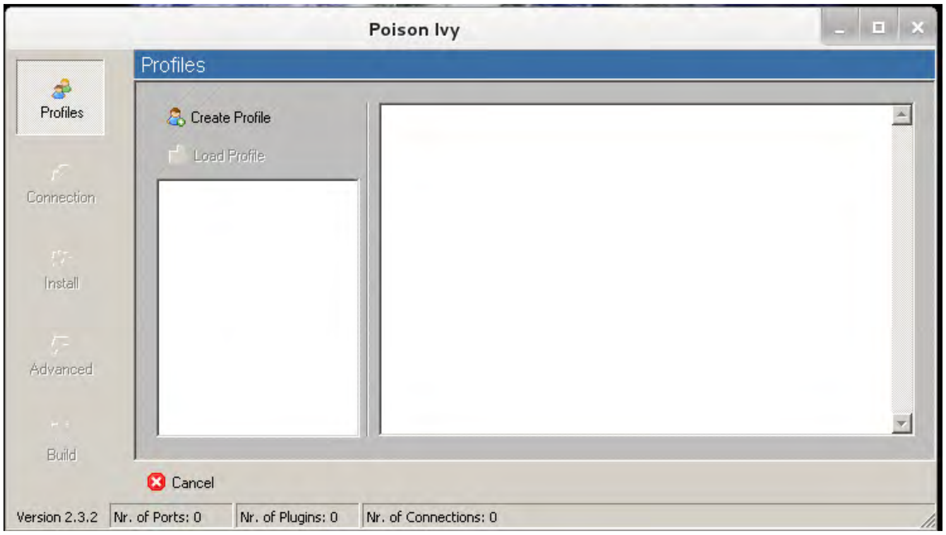
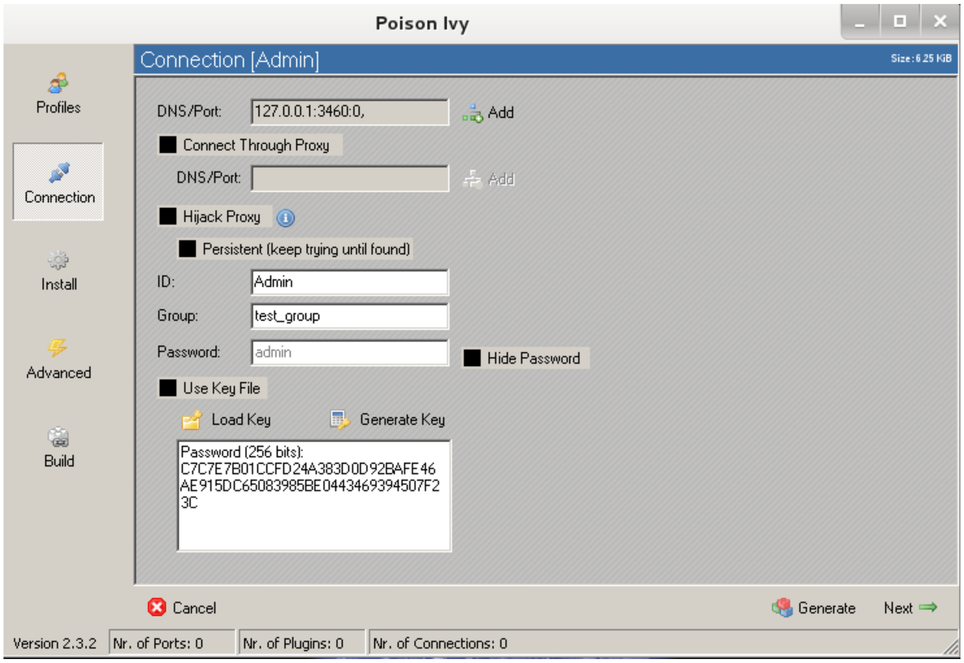


Figure 16. Poison Ivy profile window

4. Click “Add” to add the DNS/Port as shown in the connection console in Figure 17.

5. If using proxy i.e., no-ip etc

6. If using a key, generate will generate a new key or load will load an already existing key.



DNS/Port: 127.0.0.1:3460:0,

■ Connect Through Proxy

DNS/Port:

■ Hijack Proxy

■ Persistent (keep trying until found)

ID: Admin

Group: test_group

Password: admin

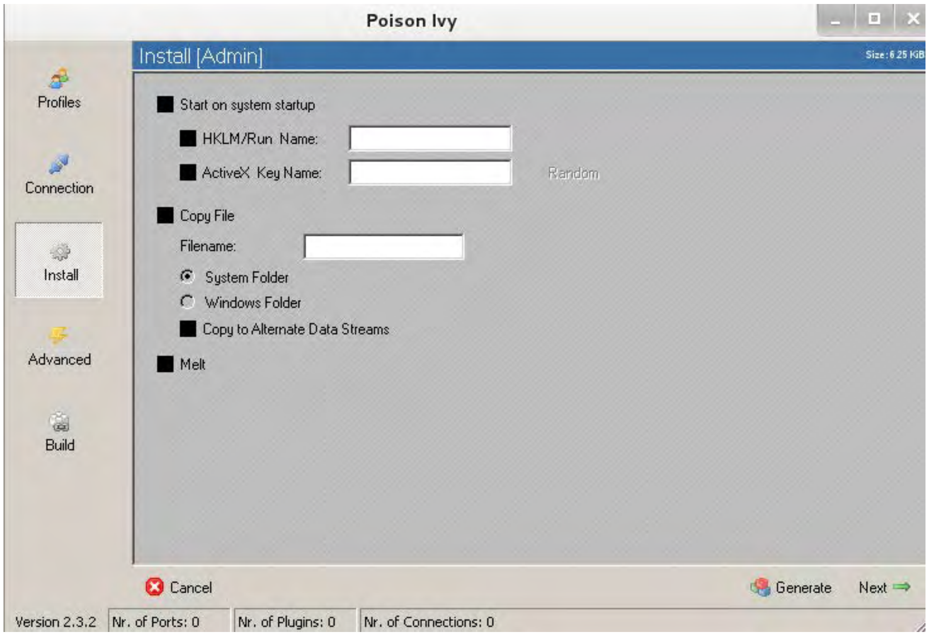
■ Hide Password

■ Use Key File

Password (256 bits):
C7C7E7B01CCFD24A383D0D92BAFE46
AE915DC65083985BE0443469394507F2
3C

Figure 17. Connection console

7. Within the install console as shown in Figure 18 provide any file name you want to use with a .exe extension



■ Start on system startup

■ HKLM/Run Name:

■ ActiveX Key Name:

■ Copy File

Filename:

● System Folder

○ Windows Folder

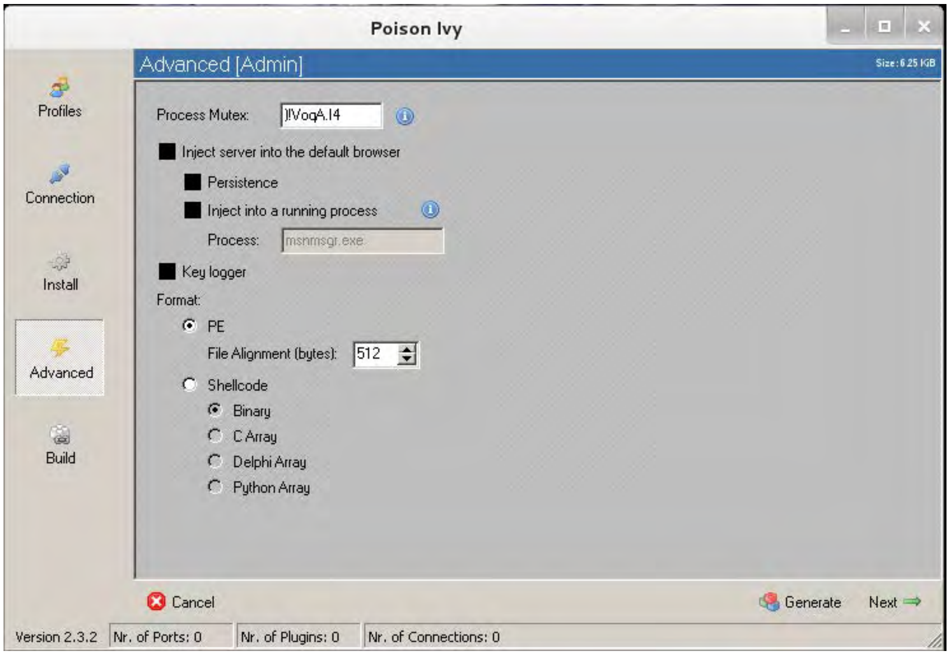
■ Copy to Alternate Data Streams

■ Melt

Figure 18. Poison Ivy install console

8. Click Icon and select the icon image/picture.

9. Click Generate to build the server file and save as shown in the Advanced Window in Figure 19.



Advanced [Admin] Size: 6.25 KB

Process Mutex:)!VoqA.I4

■ Inject server into the default browser

■ Persistence

■ Inject into a running process

Process: msnmsgr.exe

■ Key logger

Format:

● PE

File Alignment (bytes): 512

○ Shellcode

● Binary

○ C Array

○ Delphi Array

○ Python Array

Version 2.3.2 | Nr. of Ports: 0 | Nr. of Plugins: 0 | Nr. of Connections: 0

Figure 19. Poison Ivy advanced console

10. Click Ok

11. Once the server is created;

12. Give it permission using chmod 777

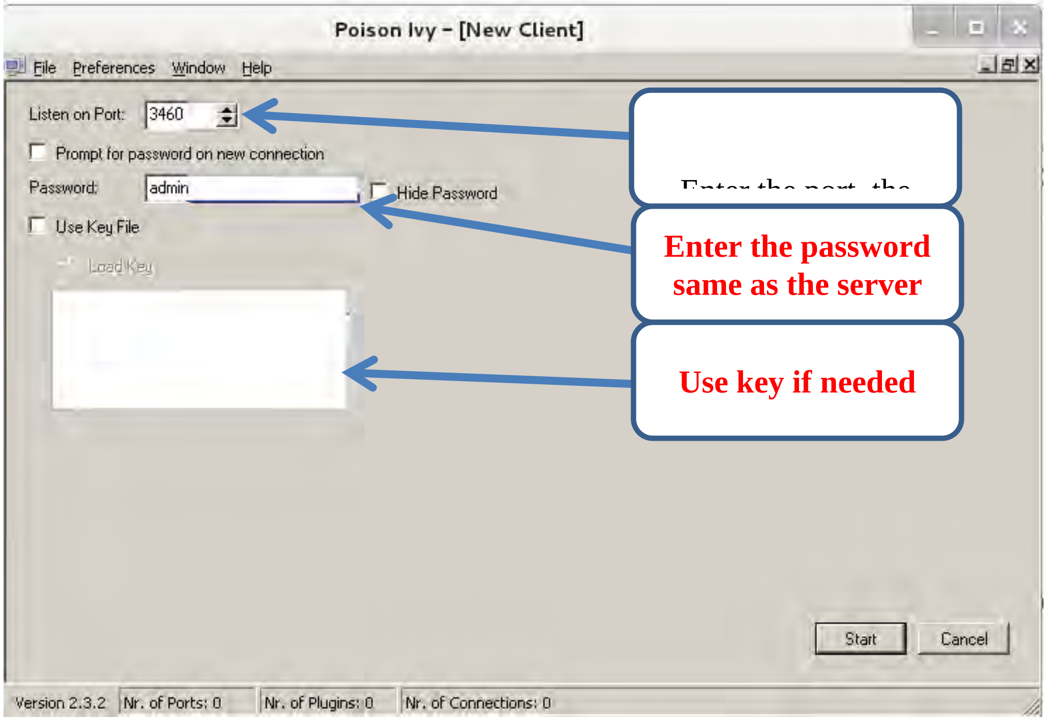
13. Set up the network i.e., open the firewall to accept connections TCP/UDP port 3460

14. Check that software firewalls allows Poison Ivy client to listen

15. Forward the necessary port

C. BUILDING POISON IVY CLIENT

1. Start the client – go to “File” – “New Client” Menu, enter the pertinent information as shown in Figure 20.



Listen on Port: 3460

☐ Prompt for password on new connection

Password: admin ☐ Hide Password

☐ Use Key File

Version 2.3.2 | Nr. of Ports: 0 | Nr. of Plugins: 0 | Nr. of Connections: 0

Enter the port, the

Enter the password same as the server

Use key if needed

Figure 20. Poison Ivy client interface

2. Click start and the system is ready to accept connections. (once connected, it will pop-up)

3. To use the server, double-click on the connection

APPENDIX E. PROPOSED LINUX TOOLSET LIST

This appendix contains a list of proposed tools to be included in a toolkit built for detecting IOCs in Linux operating systems.

```
./date

./uname -a

./who

./ps -efH

./lsof

./ifconfig -a

./netstat -anps

./netstat -an

./netstat -rn

./cat /var/lock/subsys/iptables

./ls -la /var/lock/subsys

./mount

./cat /etc/fstab

./lsmod

./pwd

./cat /root/.bash_history

chmod 744 find

./find / -printf “%m;%Ax;%AT;%Tx;%TT;%Cx;%CT;%U;%G;%s;%p\n” >
    filetimes

./chkconfig --list

./cat /etc/crontab
```

```
./ls /etc/cron.*
./cat /etc/anacrontab
./cat /etc/passwd
./cat /etc/shadow
./cat /etc/group
./cat /etc/syslog.conf
./cat /var/log/messages
./ls -lart /var/log/
./cat /var/log/messages.1
./cat /var/log/secure
./cat /var/log/cron.1
./last
./cat /var/log/secure.1
./find / -type f -xdev -exec md5sum -b {} \; > filesums
history of additional users ~username/.bash_history
./ps -ef | ./grep “lkl”
file /proc/1900/exe
./strings -a /proc/1900/exe
./hexdump -C /proc/1900/exe
file /any/linked/file
stat /any/linked/file
stat /usr/bin/.text/ircd/src/ircd
diff /usr/bin/.text/lkl/lkl /proc/1900/exe
./cat /etc/rc.d/rc.sysinit
./cat /etc/inittab
```

APPENDIX F. TOOL OUTPUT WITH NO IOCS

This appendix provides the output from each of the tools that presented no IOCs when we performed the analysis.

Table 21. Psloggedon output

```
Users logged on locally:
    8/18/2014 3:57:01 PM          ALPHA\administrator

Users logged on via resource shares:
    8/22/2014 11:11:03 AM         TRIAGE\Administrator
```

Table 22. Net session output

Baseline Output	Infected Output
Computer User name Client Type Opens Idle time --- --- \\10.1.0.2 Administrator 4 00:00:00 The command completed successfully.	Computer User name Client Type Opens Idle time --- --- \\10.1.0.2 Administrator 4 00:00:00 The command completed successfully.

Table 23. Tasklist output

Baseline Output	Infected Output
Image Name PID Session Name Session# Mem Usage ========================= ======================== System Idle Process 0 Services 0 System 4 Services 0 smss.exe 252 Services 0 csrss.exe 336 Services 0 wininit.exe 384 Services 0 csrss.exe 396 Console 1 winlogon.exe 436 Console 1 services.exe 480 Services 0 lsass.exe 488 Services 0 lsm.exe 496 Services 0	Image Name PID Session Name Session# Mem Usage ========================= ======== ================ System Idle Process 0 Services 0 24 K System 4 Services 0 640 K smss.exe 252 Services 0 532 K csrss.exe 336 Services 0 2,580 K wininit.exe 384 Services 0 2,664 K csrss.exe 396 Console 1 4,384 K winlogon.exe 436 Console 1 3,908 K services.exe 480 Services 0 6,780 K lsass.exe 492 Services 0 7,876 K lsm.exe 500 Services 0 2,480 K

```
svchost.exe        600     Services   0      svchost.exe          604     Services   0    5,408 K
svchost.exe        680     Services   0      svchost.exe          668     Services   0    4,456 K
svchost.exe        804     Services   0      svchost.exe          716     Services   0   10,000 K
svchost.exe        840     Services   0      svchost.exe          840     Services   0   33,732 K
svchost.exe        880     Services   0      svchost.exe          880     Services   0   20,308 K
svchost.exe        988     Services   0      svchost.exe          988     Services   0    8,312 K
svchost.exe        1084    Services   0      svchost.exe          1080    Services   0    9,544 K
spoolsv.exe        1172    Services   0      spoolsv.exe          1172    Services   0    7,636 K
svchost.exe        1208    Services   0      svchost.exe          1208    Services   0    6,500 K
svchost.exe        1332    Services   0      svchost.exe          1332    Services   0    5,396 K
svchost.exe        1752    Services   0      svchost.exe          1756    Services   0    3,464 K
svchost.exe        1872    Services   0      svchost.exe          1984    Services   0    5,164 K
sppsvc.exe         312     Services   0      taskhost.exe         1416    Console    1    5,240 K
SearchIndexer.exe  1428    Services   0      sppsvc.exe           2036    Services   0    6,764 K
taskhost.exe       868     Console    1      dwm.exe              1672    Console    1    3,644 K
dwm.exe            1296    Console    1      explorer.exe         1816    Console    1   40,004 K
explorer.exe       1504    Console    1      SearchIndexer.exe    972 S   Services   0    8,524 K
cmd.exe            1540    Console    1      wmpnetwk.exe         1976    Services   0    2,860 K
conhost.exe        1108    Console    1      iexplore.exe         3652    Console    1    4,368 K
iexplore.exe       3788    Console    1      PSEXESVC.exe         3852    Services   0    4,320 K
iexplore.exe       3872    Console    1      cmd.exe              312 S   Services   0    2,512 K
PSEXESVC.exe       3172    Services   0      conhost.exe          10184   Services   0    2,204 K
cmd.exe            2040    Services   0      tasklist.exe         9000    Services   0    3,904 K
conhost.exe        1188     Services  0      WmiPrvSE.exe         8836    Services   0    4,540 K
tasklist.exe       2144     Services  0
WmiPrvSE.exe       3192    Services   0
```

Table 24. Ipconfig /all output

```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : ALPHAWS01
   Primary Dns Suffix  . . . . . . . : alpha.aco
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : alpha.aco

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00–50-56-9C-73-C3
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8d90:fc57:31de:2927%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.0.25(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.1.0.1
   DHCPv6 IAID . . . . . . . . . . . : 234901590
   DHCPv6 Client DUID. . . . . . . . : 00–01-00-01-1A-D6-66-1C-00-50-56-9C-73-B7
   DNS Servers . . . . . . . . . . . : 10.1.0.21
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

```
Tunnel adapter isatap.{1575A306-ABD2-4822-997A-DE9A41818D65}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00–00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
```

Table 25.	Net share output

```
Share name   Resource                        Remark
-------------------------------------------------------------------------------
C$           C:\                         Default share
IPC$                                     Remote IPC
ADMIN$       C:\Windows                  Remote Admin
The command completed successfully.
```

Table 26.	Nbtstat –nrs output

```
Local Area Connection:
Node IpAddress: [10.1.0.25] Scope Id: []
                NetBIOS Remote Cache Name Table
Name                 Type       Host Address             Life [sec]
------------------------------------------------------------
10.1.0.2             <20>       UNIQUE 10.1.0.2          595
ALPHAWS03            <20>       UNIQUE 10.1.0.27          32
These Windows services are started:
```

Table 27.	Net start output

Baseline Scan	Infected Scan
These Windows services are started:	These Windows services are started:
Application Experience	
Base Filtering Engine	Base Filtering Engine
COM+ Event System	COM+ Event System
Computer Browser	Computer Browser
Cryptographic Services	Cryptographic Services

DCOM Server Process Launcher Desktop Window Manager Session Manager DHCP Client Diagnostic Policy Service Diagnostic Service Host Distributed Link Tracking Client DNS Client Function Discovery Resource Publication Group Policy Client IKE and AuthIP IPsec Keying Modules IP Helper IPsec Policy Agent Netlogon Network Connections Network List Service Network Location Awareness Network Store Interface Service Offline Files Plug and Play Power Print Spooler PSEXESVC Remote Procedure Call (RPC) RPC Endpoint Mapper Security Accounts Manager Security Center Server Shell Hardware Detection Software Protection SPP Notification Service Superfetch System Event Notification Service Task Scheduler TCP/IP NetBIOS Helper Themes User Profile Service Windows Audio Windows Audio Endpoint Builder Windows Defender Windows Event Log Windows Firewall Windows Management Instrumentation Windows Search Windows Time Windows Update Workstation The command completed successfully.	DCOM Server Process Launcher Desktop Window Manager Session Manager DHCP Client Diagnostic Policy Service Diagnostic Service Host Distributed Link Tracking Client DNS Client Function Discovery Resource Publication Group Policy Client IKE and AuthIP IPsec Keying Modules IP Helper IPsec Policy Agent Netlogon Network Connections Network List Service Network Location Awareness Network Store Interface Service Offline Files Plug and Play Power Print Spooler Program Compatibility Assistant Service PSEXESVC Remote Procedure Call (RPC) RPC Endpoint Mapper Security Accounts Manager Security Center Server Shell Hardware Detection Software Protection SPP Notification Service SSDP Discovery Superfetch System Event Notification Service Task Scheduler TCP/IP NetBIOS Helper Themes User Profile Service Windows Audio Windows Audio Endpoint Builder Windows Defender Windows Event Log Windows Firewall Windows Management Instrumentation Windows Media Player Network Sharing Service Windows Search Windows Time Windows Update Workstation The command completed successfully.

Table 28.   Sc query output

```
SERVICE_NAME: WMPNetworkSvc
DISPLAY_NAME: Windows Media Player Network Sharing Service
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Table 29.   Psservice output

```
SERVICE_NAME: WMPNetworkSvc
DISPLAY_NAME: Windows Media Player Network Sharing Service
Shares Windows Media Player libraries to other networked players and media devices using
Universal Plug and Play
              TYPE              : 10 WIN32_OWN_PROCESS
              STATE             : 4  RUNNING
                                     (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
              WIN32_EXIT_CODE        : 0  (0x0)
              SERVICE_EXIT_CODE : 0  (0x0)
              CHECKPOINT     : 0x0
              WAIT_HINT         : 0 ms
```

Table 30.   Driverquery output

```
Module Name    Display Name               Driver Type   Link Date
============ ====================== ============= ======================
1394ohci       1394 OHCI Compliant Ho     Kernel        7/13/2009 4:51:59 PM
ACPI           Microsoft ACPI Driver      Kernel        7/13/2009 4:11:11 PM
AcpiPmi        ACPI Power Meter Drive     Kernel        7/13/2009 4:16:36 PM
adp94xx        adp94xx                    Kernel        12/5/2008 3:59:55 PM
adpahci        adpahci                    Kernel        5/1/2007 10:29:26 AM
<…content removed for write up…>
WANARP         Remote Access IP ARP D     Kernel        7/13/2009 4:55:02 PM
Wanarpv6       Remote Access IPv6 ARP     Kernel        7/13/2009 4:55:02 PM
Wd             Wd                         Kernel        7/13/2009 4:11:31 PM
Wdf01000       Kernel Mode Driver Fra     Kernel        7/13/2009 4:11:36 PM
WfpLwf         WFP Lightweight Filter     Kernel        7/13/2009 4:53:51 PM
WIMMount       WIMMount                   File System   7/13/2009 4:17:57 PM
WmiAcpi        Microsoft Windows Mana     Kernel        7/13/2009 4:19:16 PM
ws2ifsl        Winsock IFS Driver         Kernel        7/13/2009 4:55:01 PM
WudfPf         User Mode Driver Frame     Kernel        7/13/2009 4:50:13 PM
```

Table 31. Autorunsc output

```
HKLM\System\CurrentControlSet\Services

      Microsoft Corporation

      6.1.7600.16385

      c:\windows\system32\rpcss.dll

      7/13/2009 6:09 PM
```

Table 32. Session manager registry key

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
    CriticalSectionTimeout    REG_DWORD    0x278d00
    GlobalFlag    REG_DWORD    0x0
    HeapDeCommitFreeBlockThreshold    REG_DWORD    0x0
    HeapDeCommitTotalFreeThreshold    REG_DWORD    0x0
    HeapSegmentCommit    REG_DWORD    0x0
    HeapSegmentReserve    REG_DWORD    0x0
    ProcessorControl    REG_DWORD    0x2
    ResourceTimeoutCount    REG_DWORD    0x9e340
    BootExecute    REG_MULTI_SZ    autocheck autochk *
    ExcludeFromKnownDlls    REG_MULTI_SZ
    ObjectDirectories    REG_MULTI_SZ    \Windows\0\RPC Control
    ProtectionMode    REG_DWORD    0x1
    NumberOfInitialSessions    REG_DWORD    0x2
    SetupExecute    REG_MULTI_SZ

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
AppCompatCache
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\AppPatches
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
Configuration Manager
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\DOS
Devices
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
Environment
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Executive
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
FileRenameOperations
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\I/O System
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\kernel
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
KnownDLLs
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory
Management
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Power
```

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Quota
System
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\WPA
```

Table 33. Memory management registry key

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory
Management
    ClearPageFileAtShutdown    REG_DWORD    0x0
    DisablePagingExecutive    REG_DWORD    0x0
    LargeSystemCache    REG_DWORD    0x0
    NonPagedPoolQuota    REG_DWORD    0x0
    NonPagedPoolSize    REG_DWORD    0x0
    PagedPoolQuota    REG_DWORD    0x0
    PagedPoolSize    REG_DWORD    0x0
    SecondLevelDataCache    REG_DWORD    0x0
    SessionPoolSize    REG_DWORD    0x4
    SessionViewSize    REG_DWORD    0x30
    SystemPages    REG_DWORD    0x0
    PagingFiles    REG_MULTI_SZ    ?:\pagefile.sys
    PhysicalAddressExtension    REG_DWORD    0x1
    ExistingPageFiles    REG_MULTI_SZ    \??\C:\pagefile.sys

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory
Management\PrefetchParameters
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory
Management\StoreParameters
```

Table 34. Environment registry key

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\
Environment
    ComSpec    REG_EXPAND_SZ    %SystemRoot%\system32\cmd.exe
    FP_NO_HOST_CHECK    REG_SZ    NO
    OS    REG_SZ    Windows_NT
    Path    REG_EXPAND_SZ
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\
Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
    PATHEXT    REG_SZ    .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
    PROCESSOR_ARCHITECTURE    REG_SZ    x86
    TEMP    REG_EXPAND_SZ    %SystemRoot%\TEMP
    TMP    REG_EXPAND_SZ    %SystemRoot%\TEMP
    USERNAME    REG_SZ    SYSTEM
```

```
   windir    REG_EXPAND_SZ    %SystemRoot%
   PSModulePath    REG_EXPAND_SZ    %SystemRoot%\system32\WindowsPowerShell\
v1.0\Modules\
   NUMBER_OF_PROCESSORS    REG_SZ    1
   PROCESSOR_LEVEL    REG_SZ    6
   PROCESSOR_IDENTIFIER    REG_SZ    x86 Family 6 Model 46 Stepping 6, GenuineIntel
   PROCESSOR_REVISION    REG_SZ    2e06
```

Table 35. Hivelist registry key

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Hivelist
   \REGISTRY\MACHINE\HARDWARE    REG_SZ
   \REGISTRY\MACHINE\BCD00000000    REG_SZ    \Device\HarddiskVolume1\Boot\BCD
   \REGISTRY\MACHINE\SYSTEM    REG_SZ    \Device\HarddiskVolume2\Windows\
System32\config\SYSTEM
   \REGISTRY\MACHINE\SOFTWARE    REG_SZ    \Device\HarddiskVolume2\Windows\
System32\config\SOFTWARE
   \REGISTRY\USER\.DEFAULT    REG_SZ    \Device\HarddiskVolume2\Windows\
System32\config\DEFAULT
   \REGISTRY\MACHINE\SECURITY    REG_SZ    \Device\HarddiskVolume2\Windows\
System32\config\SECURITY
   \REGISTRY\MACHINE\SAM    REG_SZ    \Device\HarddiskVolume2\Windows\System32\
config\SAM
   \REGISTRY\USER\S-1-5-20    REG_SZ    \Device\HarddiskVolume2\Windows\
ServiceProfiles\NetworkService\NTUSER.DAT
   \REGISTRY\USER\S-1-5-19    REG_SZ    \Device\HarddiskVolume2\Windows\
ServiceProfiles\LocalService\NTUSER.DAT
   \Registry\User\S-1-5-21-3079887268-1858392370-3246419219-500    REG_SZ    \Device\
HarddiskVolume2\Users\administrator\NTUSER.DAT
   \Registry\User\S-1-5-21-3079887268-1858392370-3246419219-500_Classes    REG_SZ
\Device\HarddiskVolume2\Users\administrator\AppData\Local\Microsoft\Windows\
UsrClass.dat
```

Table 36. CrashControl registry key

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\CrashControl
   AutoReboot    REG_DWORD    0x1
   CrashDumpEnabled    REG_DWORD    0x2
   Overwrite    REG_DWORD    0x1
   LogEvent    REG_DWORD    0x1
   MinidumpsCount    REG_DWORD    0x32
   DumpFile    REG_EXPAND_SZ    %SystemRoot%\MEMORY.DMP
   MinidumpDir    REG_EXPAND_SZ    %SystemRoot%\Minidump
   DumpFilters    REG_MULTI_SZ    dumpfve.sys
```

Table 37. Winlogon registry key

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
    ReportBootOk    REG_SZ    1
    Shell    REG_SZ    explorer.exe
    PreCreateKnownFolders    REG_SZ    {A520A1A4-1780-4FF6-BD18-167343C5AF16}
    Userinit    REG_SZ    C:\Windows\system32\userinit.exe,
    VMApplet    REG_SZ    SystemPropertiesPerformance.exe /pagefile
    AutoRestartShell    REG_DWORD    0x1
    Background    REG_SZ    0 0 0
    CachedLogonsCount    REG_SZ    10
    DebugServerCommand    REG_SZ    no
    ForceUnlockLogon    REG_DWORD    0x0
    LegalNoticeCaption    REG_SZ
    LegalNoticeText    REG_SZ
    PasswordExpiryWarning    REG_DWORD    0x5
    PowerdownAfterShutdown    REG_SZ    0
    ShutdownWithoutLogon    REG_SZ    0
    WinStationsDisabled    REG_SZ    0
    DisableCAD    REG_DWORD    0x0
    scremoveoption    REG_SZ    0
    ShutdownFlags    REG_DWORD    0x27
    AutoAdminLogon    REG_SZ    0
    DefaultUserName    REG_SZ    resu

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows                    NT\
CurrentVersion\Winlogon\GPExtensions
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows                    NT\
CurrentVersion\Winlogon\AutoLogonChecked
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] R. McCabe. (2014, Jun. 15). Navy, others still struggle with ditching Windows XP. [Online]. Available: http://hamptonroads.com/2014/06/navy-others-still-struggle-ditching-windows-xp

[2] G. Seffers. (2014, Aug. 20). U.S. Navy awards $2.5 billion in CANES contract. [Online]. Available: http://www.afcea.org/content/?q=node/13340

[3] McAfee. (2013, Apr. 1). Infographic: The State of Malware 2013. [Online]. Available: http://www.mcafee.com/us/security-awareness/articles/state-of-malware-2013.aspx

[4] C. Rodriguez. "Defending Against Increasingly Sophisticated Cyber Attacks HP TippingPoint Bolsters Enterprise Data Center Protection," Frost and Sullivan, Washington, DC. Nov. 2013.

[5] ENISA. Triage. [Online]. Available: https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases/triage-1

[6] Redline v1.11 User Guide. Mandiant Corp. Alexandria, VA. 2013, pp. 5–51.

[7] A. Moser and M. Cohen, "Hunting in the enterprise: Forensic triage and incident response," *Digital Investigation*, vol. 10, no. 2, pp. 89–98, Sep. 2013.

[8] Mandiant Intelligent Response. Mandiant Corp. Alexandria, VA. 2013, p 2.

[9] Palo Alto Networks. APT Prevention. [Online]. Available: https://www.paloaltonetworks.com/products/features/apt-prevention.html

[10] Palo Alto Networks. Palo Alto Networks: WildFire datasheet. [Online]. Available: https://www.paloaltonetworks.com/content/dam/paloaltonetworks-com/en_US/assets/pdf/datasheets/wildfire/wildfire.pdf

[11] M. Murphy, A. LeMasters. RETRI: Rapid enterprise triaging. [Online]. Available: http://www.blackhat.com/presentations/bh-usa-09/LEMASTERS/BHUSA09-LeMasters-RETRI-SLIDES.pdf

[12] M. Murphy, A. LeMasters. (2009, Jun 1). Rapid Enterprise Triaging (RETRI): How to Run a Compromised Network and Keep Your Data Safe. [Online]. Available:https://www.blackhat.com/presentations/bhusa09/LEMASTERS/BHUSA09-LeMasters-RETRI-PAPER.pdf

[13] GNS3. (2014, Jul. 2). Switching Simulation in GNS3. [Online]. Available: http://www.gns3.net/documentation/gns3/switching-simulation-in-gns3/

[14] DD-WRT. (2014, Jun. 23). X86 release 24461. [Online]. Available: ftp://ftp.dd-wrt.com/others/eko/BrainSlayer-V24-preSP2/2014/06-23-2014-r24461/x86/

[15] Cisco. (2006, Aug. 25). Inter-Switch Link and IEEE 802.1Q Frame Format. [Online]. Available: http://www.cisco.com/c/en/us/support/docs/lan-switching/8021q/17056-741-4.html

[16] Microsoft. (2012, Apr. 17). Technet: Date. [Online]. Available: http://technet.microsoft.com/en-us/library/cc732776.aspx

[17] Microsoft. Technet: Time. [Online]. Available: http://technet.microsoft.com/en-us/library/bb491015.aspx

[18] Microsoft. (2010, May 24). Technet: The ipconfig.exe command line tools. [Online]. Available: http://technet.microsoft.com/en-us/library/ee624064(v=ws.10).aspx

[19] Microsoft. Technet: Net sessions. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490711.aspx

[20] Microsoft. Windows XP Professional Product Documentation: net use. [Online]. Available: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/net_use.mspx?mfr=true

[21] Microsoft. Technet: Net share. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490712.aspx

[22] Microsoft. Technet: Net start. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490713.aspx

[23] Microsoft. (2012, Apr. 17). Technet: Sc query. [Online]. Available: http://technet.microsoft.com/en-us/library/dd228922.aspx

[24] Microsoft. Technet: Driverquery. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490896.aspx

[25] Microsoft. Technet: Tasklist. [Online]. Available: http://technet.microsoft.com/en-us/library/bb491010.aspx

[26] Microsoft. Technet: Netstat. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490947.aspx

[27] Microsoft. Technet: Nbtstat. [Online]. Available: http://technet.microsoft.com/en-us/library/cc940106.aspx

[28] Microsoft. (2012, Apr. 17). Technet: Reg query. [Online]. Available: http://technet.microsoft.com/en-us/library/cc742028.aspx

[29] Microsoft. Technet: Doskey. [Online]. Available: http://technet.microsoft.com/en-us/library/bb490894.aspx

[30] M. Russonovich. (2014, May 2). Windows sysinternals: Psexec v2.11. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx

[31] M. Russonovich. (2010, Apr. 28). Windows sysinternals: Psservice v2.24. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897542.aspx

[32] M. Russonovich. (2010, Apr. 28). Windows sysinternals: Psloggedon v1.34. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897545.aspx

[33] M. Russonovich. (2010, May 6). Windows sysinternals: Logonsessions v1.21. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb896769.aspx

[34] M. Russonovich. (2012, Mar. 23). Windows sysinternals: Pslist v1.3. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb896682.aspx

[35] M. Russonovich. (2013, Jan. 24). Windows sysinternals: Handle v3.51. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb896655.aspx

[36] M. Russonovich. (2014, Aug. 18). Windows sysinternals: Autoruns for windows v12.02. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

[37] M. Russonovich. (2011, Jul. 18). Windows sysinternals: Listdlls v3.1. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb896656.aspx

[38] M. Russinovich. (2014, May 2). Windows sysinternals sigcheck v2.1. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897441.aspx

[39] Microsoft. MUI Resource Management. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/dd319070(v=vs.85).aspx

[40] K. Seng. (2011, Mar 2). Windows Sysinternals forum. *Authenticate tool copies*. [Online]. Available: http://forum.sysinternals.com/authenticate-tool-copies_topic25212.html

[41] J. Leyden. (2013, Aug. 27). Poison ivy rat becoming the ak-47 of cyber-espionage attacks. [Online]. Available: http://www.theregister.co.uk/2013/08/27/poison_ivy_rat_apt/

[42] J. Bennet et al. (2013, Aug. 21). Poison Ivy: Assessing damage and extracting intelligence. [Online]. Available: http://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf

[43] Trend Micro. Threat encyclopedia: Poisonivy. [Online]. Available: http://about-threats.trendmicro.com/apac/malware/poisonivy

[44] D. Parker. (2005, Feb. 16). Windows NTFS alternate data streams. [Online]. Available: http://www.symantec.com/connect/articles/windows-ntfs-alternate-data-streams

[45] Kali. (2013, Feb. 25). What is Kali Linux? [Online]. Available: http://docs.kali.org/introduction/what-is-kali-linux

[46] H. Carvey and E. Casey, Windows Forensic Analysis. Burlington, MA: Syngress, 2009, pp. 18–34.

[47] Microsoft. NetBIOS Suffixes (2014, 08, 24). [Online]. Available: http://support.microsoft.com/kb/163409

[48] Microsoft. (2014, Aug, 30). Registry Hives. [Online] Available: http://msdn.microsoft.com/en-us/library/windows/desktop/ms724877(v=vs.85).aspx

[49] B. Carrier, File System Forensic Analysis. Upper Saddle River, NJ: Addison-Wesley, 2005, pp 316–320.

[50] G. Edwards. (2012, Jan. 3). Windows Timestamp Tampering. [Online]. Available: http://blog.opensecurityresearch.com/2012/01/windows-timestamp-tampering.html

[51] D. Hull, (2010, Nov. 2). Digital forensics: Detecting time stamp manipulation. [Online]. Available: https://blogs.sans.org/computer-forensics/files/2010/10/ts_change_rules_gui1.jpg

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California